Call-A.P.P.L.E.™

Volume V, Number 12 December 1982 $2.50 ($3.00 Canada, £1.75 UK.)

A Call —A.P.P.L.E. TUTORIAL:
Basic Suggestions
Page 9

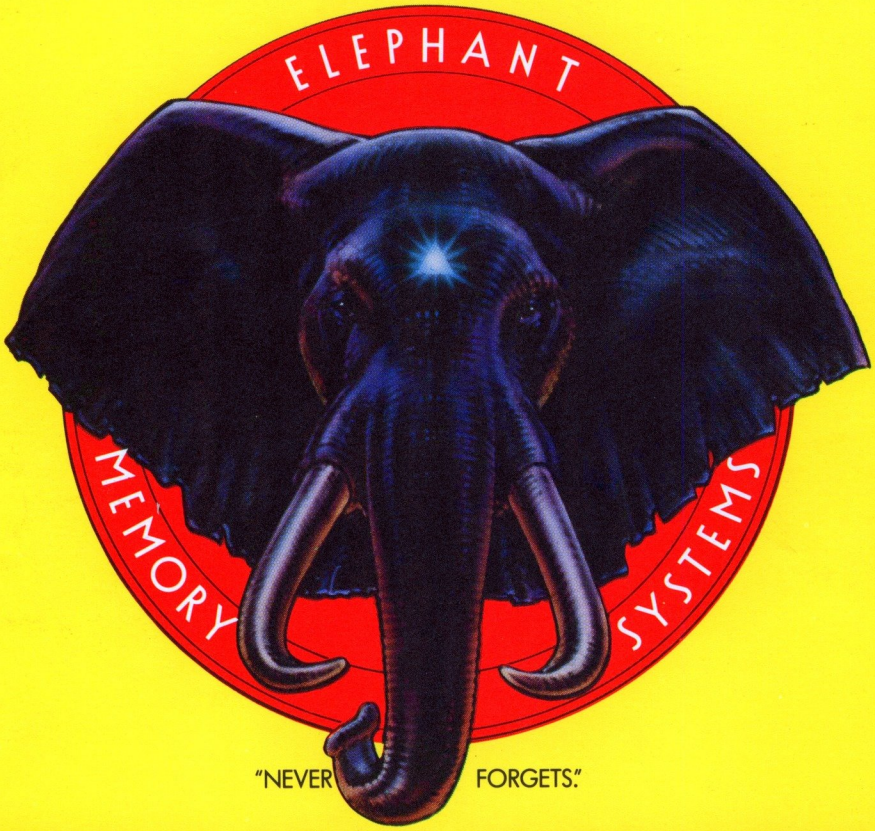
A Call —A.P.P.L.E. TUTORIAL:
Doing It With A Naked Apple
Page 15

FROM THE VERY CORE OF APPLE:
The Unknown INTFLG
Page 23

EXPLORING THE WORLD OF APPLE:
New Uses For The RAM Card
Page 25

Apple PugetSound Program Library Exchange

REMEMBER:



"NEVER FORGETS."

MORE THAN JUST ANOTHER PRETTY FACE.

Says who? Says ANSI.

Specifically, subcommittee X3B8 of the American National Standards Institute (ANSI) says so. The fact is all Elephant™ floppies meet or exceed the specs required to meet or exceed all their standards.

But just who is "subcommittee X3B8" to issue such pronouncements?

They're a group of people representing a large, well-balanced cross section of disciplines—from academia, government agencies, and the computer industry. People from places like IBM, Hewlett-Packard, 3M, Lawrence Livermore Labs, The U.S. Department of Defense, Honeywell and The Association of Computer Programmers and Analysts. In short, it's a bunch of high-caliber nitpickers whose mission, it seems, in order to make better disks for consumers, is also to make life miserable for everyone in the disk-making business.

How? By gathering together periodically (often, one suspects, under the full moon) to concoct more and more rules to increase the quality of flexible disks. Their most recent rule book runs over 20 single-spaced pages—listing, and insisting upon—hundreds upon hundreds of standards a disk must meet in order to be blessed by ANSI. (And thereby be taken seriously by people who take disks seriously.)

In fact, if you'd like a copy of this formidable document, for free, just let us know and we'll send you one. Because once you know what it takes to make an Elephant for ANSI . . .

We think you'll want us to make some Elephants for you.

ELEPHANT.™ HEAVY DUTY DISKS.

For a free poster-size portrait of our powerful pachyderm, please write us.
Distributed Exclusively by **Leading Edge Products, Inc.**, 225 Turnpike Street, Canton, Massachusetts 02021
Call: toll-free 1-800-343-6833; or in Massachusetts call collect (617) 828-8150. Telex 951-624.

Nothing like it before. Nothing else like it now!
. . . brings you continuous Hi-Res action-animation in every adventurous moment! And, real running, leaping, crawling. Real fighting, shooting, stabbing, dynamiting. Real wounding, poisoning, killing. Real action, excitement, mystery! All in a real-time challenging adventure that's the wave of the future!
Paul Stevenson's graphic genius, first displayed in his best selling "Swashbuckler" sword fighting game, outdoes itself in AZTEC. You're inside an ancient Aztec pyramid searching for the golden idol. Descend deep into the heart of the temple—meet cobras, scorpions, giant lizards, hostile Aztec guardians and more. Watch for hidden trapdoors and strange death-rooms. Be ready to fight, or run, crawl or jump to possible safety. The menace is real, the options and strategy are yours. You've never seen an adventure like Aztec! You'll never tire of its amazing action-animation and exciting challenge. $39.95 for the Apple II* At your computer store or:
DATAMOST
9748 Cozycroft Ave., Chatsworth, Ca 91311. (213) 709-1202.
AZTEC
VISA/MASTERCARD accepted. $2.00 shipping/handling charge. (California residents add 6½% sales tax.)
*Apple II is a trademark of Apple Computer, Inc.

APPLE PUGETSOUND PROGRAM LIBRARY EXCHANGE:

CLUB INFORMATION

GENERAL OFFICES
304 Main Ave. S., Suite 300
Renton, WA 98055 — (206) 271-4514
Hours: 8 - 4 Mon. - Fri.
Member Services/Dealer Orders

EDITORIAL DEPARTMENT
5606 California Ave. S.W.
Seattle, WA 98136—(206)932-8288
Hours: 8-4 Monday - Friday

ADVERTISING REPRESENTATIVES
Grawin Publications
1020 Lloyd Building
Seattle, WA 98101 (206) 223-0861

MEETING NOTICES

7:30 PM, Tuesday, December 21, 1982

Swan Computers
1034 - 116th Ave. N.E., Bellevue

Guest Speaker
Jay Bakst/Alternative to Flow Charting

ANNUAL NOMINATIONS

7:30 PM, Tuesday, January 18, 1983

Red Lion Inn
18740 Pacific Hiway S.

Guest Speaker
Mike DeVille/Basis 108 Computer

7:30 PM, Tuesday, February 15, 1983

Speaker and location to be announced

CALL —A.P.P.L.E. HOT LINE

Technical/programming assistance only
(206) 932-6588
Hours: 9 AM - 3 PM and
6 PM - 10 PM 7 days per week
Pacific Time

Please refer Pascal, Apple ///, VisiCalc, questions directly to consultants on page 95.
See page 85 of the June '82 issue for information on use of the Hotline.
Please also check our list of consultants on page 95 for a specialist in your field.

Membership Information

Please direct telephone questions on membership and orders to (206) 271-4514.

One Time Apple—Cation Fee	$25.00
Annual Dues	$20.00
Total	$45.00

Additional postage required for foreign memberships and shipments.

Additional membership postage:

- Canada, West Indies, Central America, Columbia, Venezuela and Mexico — Add $15.00 ($60.00 total)
- All S. American countries (except above), Europe and N. Africa — Add $21.00 ($66.00 total)
- Asia, Mid East, S. Africa, Australia and Pacific Islands — Add $27.00 ($72.00 total)

All orders must be prepaid in U.S. funds only.

Membership is on an annual, 12 month basis, commencing with the month of enrollment. Upon receipt of total membership fees, new members will receive an Apple—Cation blank, order form and description sheet of club products.

Please allow up to 30 days for a turnaround on membership and orders.

Call —A.P.P.L.E. is mailed free twelve time per year as benefit of membership. Subscriptions are not available.

1982 back issues available at $2.50 each.

Call —A.P.P.L.E. bound volumes for previous years are available as indicated on the member order form on the last page.

Institutional memberships are now available. Contact General Office for details.

Prospective authors please send self-addressed stamped envelope (37¢ postage) to Editorial Dept. for a copy of Author's guidelines.

Cover and interior design by Vic Warren Design.

NOTICE

It is essential that membership number and expiration date be included on all orders and correspondence. This information is found on the top line of the mailing label of each issue of **Call —A.P.P.L.E.**

Please be aware of your expiration date so you can renew in time to avoid missing any issues of **Call —A.P.P.L.E.** Renewal notices are sent via first class mail approximately two months prior to the expiration date. Please check your own expiration date in the event your notice is not delivered (there has been a slight problem with delivery).

Changes of address must be received at the General Office at least 30 days prior to effective date of move. Issues missed due to non-receipt of change of address notification must be separately reordered by the member at current cost. Change of address form may be found on page 95 of this issue.

A.P.P.L.E. CRATE BULLETIN BOARD
(206) 935-9119
(usually 24 hours)

Officers and Directors

Dick Hubert, President	(206) 271-6939
Bob Huelsdonk, Vice President	(206) 362-4910
Dave Lingwood, Secretary	(206) 244-9360
Bob Clardy, Treasurer	(206) 641-1917
Val J. Golding, Director	(206) 932-6588
Ken Smith, Director	(206) 322-4199
Fred Merchant, Director	(415) 921-7792

Office Staff

Dick Hubert	Executive Director
Dave O'Leary	Administrative Assistant
Sally Davies	Admin. of Information Systems
Arliss Sword	Dealer Orders
Patti Hoffmann	Special Projects Coordinator
Gunter Hirt	Bookkeeping, Member Records
Joff Morgan	Technical Operations
Cheryle McCalmant	Order Processing
Kris Mortenson	Order Processing
Sharon Birdsell	Order Processing
Gayle Rank	Secretary/Receptionist
Mike Christensen	Systems Development
Carol Jackson	Shipping/Purchasing
Brad Phillips	Shipping/Receiving
Steve Johnson	Ap·tronics
Steve Ross	New Product Development
Garry Reinhardt	Systems Development

Magazine Staff

Val J. Golding	Editor	(206) 932-8288
Kathryn Hallgrimson	Office Administrator	(206) 932-8288
Kathy Cherry	Editorial Assistant	(206) 932-8288
Jonathan V. Post	Contributing Editor	(206) 824-7641
Ron DeGroat	Pascal Editor	(303) 443-2722
Kevin O. Donohoe	Production Editor	(206) 223-0861
Jim Hockenhull	Staff Writer	(509) 334-4137
Paul Mosher, M.D.	Staff Writer	(518) 765-4118
Lee Reynolds	Staff Writer	(305) 722-2639
Charles Rusch	Staff Writer	(503) 484-4618
David G. Sparks	Staff Writer	(319) 232-2313
Ralph H. Swerdlow, M.D.	Staff Writer	(916) 920-8709
Grawin Publications	Production, Advertising	(206) 223-0861

A.P.P.L.E. consultants will be found on page 95 of this issue.

Call-A.P.P.L.E.™

Volume 5, Number 12 December 1982/Menu Val J. Golding, Editor

ENTRÉES

9 A CALL -A.P.P.L.E. TUTORIAL:
Basic Suggestions
Luis M. Hernandez

15 A CALL -A.P.P.L.E. TUTORIAL:
Doing It With A Naked Apple
Relocating Machine Language
Bill Starbuck

23 FROM THE VERY CORE OF APPLE:
The Unknown INTFLG
Kelly W. Puckett

25 EXPLORING THE WORLD OF APPLE:
New Uses for the RAM Card
Mark Harris

À LA CARTE

31 **Converting Applewriter Files**
Thomas G. Williams

37 **Video Follow Up**
Jim Hockenhull

39 **Applemash**
Michael Thyng

43 **NEXT**
Wes Huntress

45 **Disk Mapping the Z-80 CP/M System**
Art Messler

49 **Programming and Problem Solving**
Dr. David A. Lingwood

55 **Extra, Extra**
Craig Peterson

57 GET MOST WRAP UP:
Yes, You Really Can Put Files on Track 0
Clay Ruth

63 **Christmas Present From a Hacker**
Phillip Chien

65 A IS FOR A.P.P.L.E.
Print ASCII
Ralph Swerdlow/Call -A.P.P.L.E. Staff Writer

67 A CALL -A.P.P.L.E. REVIEW:
Pascal From BASIC
David Sparks/Reviewer

71 **Ctrl-C Killer**
S. Scott Zimmerman

HORS d'OEUVRES

5 The Editor Bytes Back
The Editor

7 Bytes From A.P.P.L.E.
Val J. Golding

5, 37, 39, 43, 46, 55, 59, 66, 70, 71, 77, 80, 82
So What Did You Expect?

75 Robotics Series
Michael Thyng

77 Our Biggest Contest Yet!

78 Our Smallest Contest Yet!

79 A.P.P.L.E. Doctor
Cornelis Bongers/Call -A.P.P.L.E. Staff Writer

82 Write —A.P.P.L.E.
The Readers

83 Tommorrow's Apples Today
Kevin O. Donohoe/Editor

95 Advertiser's Index

95 Call -A.P.P.L.E. Consultants

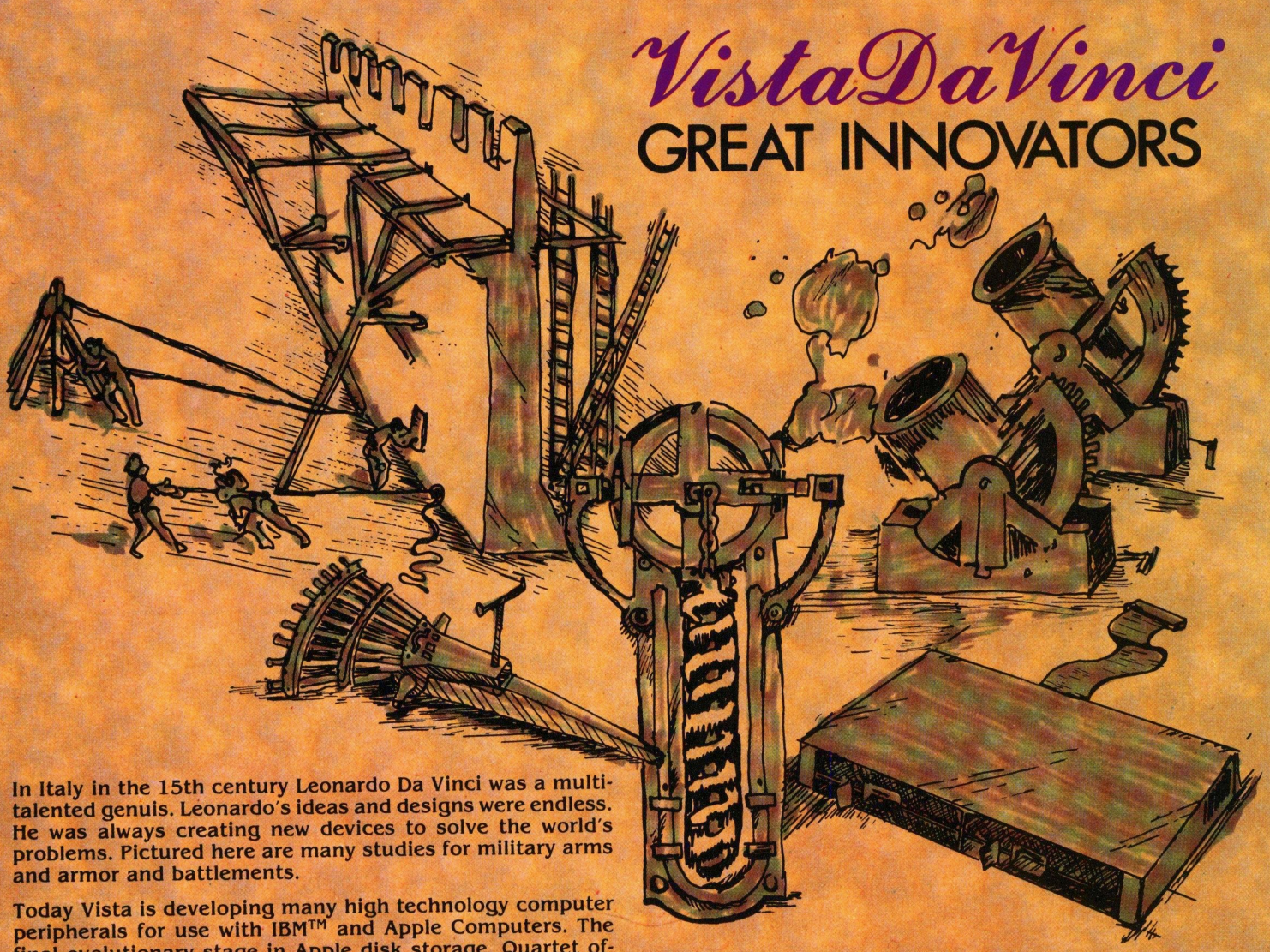
95 A.P.P.L.E. Order Form

96 Call -A.P.P.L.E. Unclassifieds

Apple user groups may reprint without prior permission any portion of the contents herein, provided author, title, and publication credits are given and a written notice of such use furnished to the editorial office. For commercial applications of programs herein, the author should be contacted c/o General Office. Call —A.P.P.L.E. is published monthly by Apple Puget Sound Program Library Exchange, 304 Main Ave. S., Renton, WA 98055. Subscriptions are not available but copies are mailed free as a benefit of A.P.P.L.E. membership.

Entire contents copyright© 1982 by
Apple PugetSound Program Library Exchange

Vista DaVinci
GREAT INNOVATORS



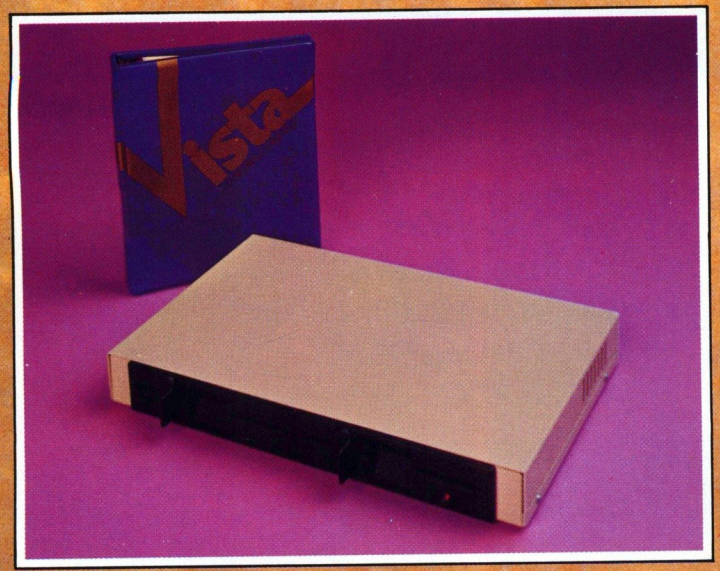
In Italy in the 15th century Leonardo Da Vinci was a multi-talented genuis. Leonardo's ideas and designs were endless. He was always creating new devices to solve the world's problems. Pictured here are many studies for military arms and armor and battlements.

Today Vista is developing many high technology computer peripherals for use with IBM™ and Apple Computers. The final evolutionary stage in Apple disk storage, Quartet offers you the capacity of 4 Apple Disk Drives in the volume of one. Quartet's low profile styling has been developed to blend in with the profile and style of your Apple II Computer.

Quartet uses 2 Double Sided Thinline disk drives to give you over 640K of storage. With the included Quartet Controller & Software, your Computer can handle much larger tasks than before. When using protected software, the Quartet operates just like 2 Apple Disk Drives, and will operate with ALL Apple compatible software.

The Quartet system is the COMPLETE, one-stop solution to your storage requirements on the Apple or Apple Compatible Computer. No other two-drive package offers as much value at any price. And the Quartet offers all this with a price comparable to single drive, single sided systems.

Features: • Thinline Drives for low profile • 2 Double Sided Drives for 640K Storage • Quartet Controller for Single/Double Sided Operation • Emulates Apple Disk II under a single sided mode • Boots & Runs All Apple Software • Double Sided, 40 track Patch software for DOS, CP/M*, & Pascal • Full Vista 120 Day Warranty



Vista Quartet

Contact Your Local Vista Dealer or Call our Vista Hotlines.

Vista COMPUTER COMPANY, INC.

1317 East Edinger / Santa Ana, CA 92705
(714) 953-0523 / (800) 854-8017

DISTRIBUTORS / REPRESENTATIVES

Western - Group 3 Wholesale
(213) 973-7844 (408) 732-1307
South Central - M.P. Systems
(214) 385-8885
UK - Hal Computers Ltd.
(0252) 517175 / TWX: 858404
Northeast - Computers & Peripherals Inc.
(315) 476-6664
Florida - Audio Marketing (REP)
(305) 322-8327
Central - Wyatt & Associates
(317) 773-4791

*CP/M is a trademark of Digital Research, Inc.

TM IBM is a registered trademark of International Business Machines.

THE EDITOR BYTES BACK

the editor

VIEW from 37,000 feet: enroute to Houston Applefest; nearly clear skies, only a hint of haze, scattered tan-tinged cloudlets from the corner of the eye. One of the fleeting moments of relaxation we get, an opportunity to think ahead to the new year, a glimpse into the **Call -A.P.P.L.E.** crystal ball.

What's in store for the year 1983? When rumors circulate so widely the furthest outreaches know of them, rumor becomes fact. In true **Call —A.P.P.L.E.** tradition, we plan in depth coverage of the newly redesigned Apple][motherboard as soon as it is announced along with appropriate tech notes when they become available. **Call —A.P.P.L.E.** will tell you first about other new Apple Computer, Inc., products as they are announced.

RAM card prices are dropping to record lows; our own **AP.RAM/16** is only one such example. Starting in this issue, we'll help you explore the untapped potential of a RAM card. **Bill Starbuck's Doing it With a Naked Apple** suggests some possibilities; others come to light in Part 1 of **New Uses for the RAM Card** which introduces **Mark Harris** to our pages, along with a new concept: a continued story, continued not from one issue to the next, but from one magazine to another! Part 2 which explains how to use the RAM card as a copy program buffer, appears in issue no.3 of our companion publication **Call -A.P.P.L.E. in Depth: All About DOS.** Sure, this is another of our "sneakys," but when you get it, you'll find that like its predecessors, it will rapidly become one of your most valued reference works.

In almost every issue of **Call -A.P.P.L.E.** we introduce authors who have sought us out with a new technique to reveal, or solid information to assist new Apple users. In this issue, all four cover stories fall into that category, thus it behooves us to remind our readers, members and non-members, that we are interested in what you have to offer. We are not hyper-critical on style; that is what editors are paid to do. If you have a technique or application you would like to share with your fellow members, send a 37 cent stamped, self-addressed envelope to the Editorial office as listed on Page 2, and ask for a copy of our authors' guidelines. We have a minimum rate of $50 per published page.

One of the things we pride ourselves on is the ability to drop in, virtually at press time, an entire new story. This is what happened at Houston Applefest when we met author **Kelly Puckett.** We talked for a spell then, while he was having a byte to eat back at the concession area, Kelly cranked out **The Unknown INTFLAG** in a matter of moments, along with a short demo program. If you've been looking for a way to distinguish variable types that you are passing to an assembly language program, then you'll want to read this note.

From the Phillipines comes our fourth feature story by first-timer **Luis Hernandez: Basic Suggestions for a Novice Programmer.** Here Luis puts in print some basic facts and techniques often overlooked by the beginner. We suggest you take a look.

Also of interest, **A.P.P.L.E.** secretary **Dave Lingwood** explores **Problem Solving in BASIC,** and offers some hints on avoiding the infamous Applesoft rounding error, as does staff writer **Cornelis Bongers** in this issue's **A.P.P.L.E. Doctor.**

Last month **Dave Sparks** presented persuasive evidence of how easy assembly language really is along with a convincing example of directly converting a BASIC program, line by line, to assembly. This time around, we have a short feature entitled **Apple Writer Converter** by **Thomas G. Williams** and ourselves which converts an Applewriter I binary file to a standard ASCII text or binary file. It is presented first as a pure BASIC program, then as a combination BASIC and assembly program. Try both versions yourself; you'll find the speed differences are **dramatic.** This is just one reason why we constantly preach and publish assembly language.

One utility worthy of special mention is **Ctrl-C Killer.** Author **S. Scott Zimmerman** has his own (explained) reason for having written it, but one important and unmentioned idea is its application in making a program "user proof". Take a glance; you'll see what we mean.

This is the final month for the publication of our **So What Did You Expect Dept.** contest entries; you'll find them scattered throughout this issue, along with details of the **So What** contest and **Michael Thyng's Eamon Adventures** contest.

We note with pride this is our biggest **Call -A.P.P.L.E.** to date. We think it's all exciting reading, and there's still a bunch more on the contents page.

See if you don't agree.

```
]LIST

5  REM

SO WHAT DID YOU EXPECT?

MICHAEL ZIOBER : CALL
      -A.P.P.L.E.

10 POKE 54,47: POKE
     55,251: PRINT
     : CALL 65171:
     POKE 56,88: POKE
     57,252: GET A$:
     CALL 65161: POKE
     37,11: CALL 64546:
     POKE 36,8
20 READ A: IF A > 0
     THEN  POKE  PEEK
     (40) +  PEEK (41)
     * 256 +  PEEK
     (36),A: CALL 64500:
     GOTO 20
30 CALL 976: DATA
     211,207,160,215,200,19
     3,212,160,196,201,196,
     160,217,207,213,160,19
     7,216,208,197,195,212,
     191,0
```

CO-PROCESSING

The hardware answer to software portability. From PCPI The COPROCESSING™ Company.

Now you can run all the most popular CP/M® packages on your Apple, with APPLI-CARD™, the one-card answer to CP/M portability!

Only APPLI-CARD provides you with CP/M *plus all these features*:

- One card WordStar® execution with upper and lower case
- 64K on-card memory
- 40-255 column horizontal scroll
- Choice of application
- 2K PROM, Realtime clock and expansion interface all on-card
- Z80A® or Z80B with memory
- Menu driven set up
- 63K available for program development or execution

$1100 value for $595

BUY THIS — 6MHz APPLI-CARD (with all the features described above) for $595 **—AND GET THIS TOO** FMS-81 Data Management System!

This is the exact same top-rated $495 Data Management System you've seen reviewed in leading computer publications. And it can be yours by return mail when you purchase your 6MHz APPLI-CARD from your local dealer. Simply mail us the special enclosed warranty card and $14.95 to cover handling and shipping charges and FMS/81 is yours!

$545 value for $445

BUY THIS — 4MHz APPLI-CARD (with all the features described above) for $445 **—AND GET THIS TOO** TARGET PlannerCalc!

PlannerCalc is the new, best selling $99 financial planning electronic spread sheet. And it can be yours by return mail when you purchase your 4MHz APPLI-CARD from your local dealer. Simply mail us the special enclosed warranty card and $14.95 to cover handling and shipping and PlannerCalc is yours!

THAT'S NOT ALL!

WordStar, CBasic, dBASEII and other CP/M-based programs are available with APPLI-CARD at low package-prices! Contact your local dealer.

(Dealers may contact SOFTSEL, MICRO D, Lifeboat Associates or other selected distributors for information.)
If you prefer contact us directly for prices and your nearby dealer where you can see APPLI-CARD in action.

PCPI

Personal Computer Products, Inc.
16776 Bernardo Center Drive, San Diego, California 92128 (619) 485-8411
The COPROCESSING™ Company

Co-Processing is a trademark of Personal Computer Products, Inc. APPLI-CARD is a trademark of Personal Computer Products Inc.
CP/M is a registered trademark of Digital Research, Inc. WordStar is a registered trademark of Micropro, Inc.
Z80 is a registered trademark of Zilog, Inc.

BYTES FROM A.P.P.L.E.

Val J. Golding

GOOD Morning. Welcome to the Christmas season and all that goes with it. By now we hope you have received our 1982 Christmas wish list. If not, don't call... we've either fallen behind schedule or Uncle Sam and the postal service have. Our catalog has gone to you via first class mail, never fear.

One of the interesting things about attending Applefest and computer shows is some of the "deals" we can make on the spot to save our members some cash on publications and hardware. F'rinstance: *Bill Sanders,* a contributor to *Call –A.P.P.L.E.* in the past, has written a book **The Elementary Apple.** We spent some time looking at it, found it a very instructive tutorial, and said "Let's do it!," so here it is at $11, a savings of over 25%.

F'rinstance number two is our new hobby/prototyping board, fully drilled and tinned, and with gold plated edge connectors, just waiting for your components at $15. Next comes our **Ap.RAM/16,** *modestly* priced at $50, ready for you to load Big Mac.LC, Integer Basic, DOS or your pleasure. This ties in neatly with a couple of articles in this and forthcoming issues that say: "now you've got it, here's what it can do".

Here's another one, fresh from the 'fest. We've been looking for some time now for a quality joystick that met our standards, and we think we've found it. Watch this space next month for details on this and a sensational low-priced copy holder.

One hardware product which took us almost completely off guard is an innovative item undescriptively called the **LCM100.** What LCM stands for is *Line Carrier Modem,* a modem that works not on telephone lines, but on household AC ciruits, at a rate of up to 9600 baud. With the LCM100 you can connect a printer or video terminal/keyboard combination up to 800 feet away from your Apple and control the Apple from the keyboard or output to the printer. The retail price on this is $365; members of A.P.P.L.E. can steal it for $225!

Publications, hardware, what else? Oh yes, software, can't forget that! **Ap. Graphics: the Filler** is the first in a series of programmers' graphics utilities. The Filler is a fast, fast, fast assembly program where you can draw an outline and almost instantaneously fill it with any one of 21 available colors. For less than 25 bucks you can't go wrong. Available early 1983.

Do we support the Apple *III* we're often asked. Yes, indeed we do! **Word Weaver***III* was among the first, and among the finest word processors for the Apple *III*; a special sale price of just $60 is one of our Christmas gifts for you. Between now and January 31st, you can save $15 by ordering it.

Diskettes with programs on them are called "software," but what do you call them if they are blank? Certainly not "hardware." Media perhaps? That does seem to fit the bill. In any case, pick up on our diskette special, good through January 31st, for ten CDC label diskettes in a reuseable fiberboard box for only $23. No limit.

We wish we could tell you more but you'll just have to wait another month. One thing you can be assured of: we'll almost certainly turn out a new quality product or three from the San Francisco Applefest.

BOOT

Craig MacIntosh

Crown Syndicate, Inc., Five Crown Road, Weatherford, Texas 76086. Telephone (817) 594-5125.

EVERYONE NEEDS A . . .

UNINTERRUPTABLE POWER SOURCE

ADVANCED DESIGN
NEW RELIABILITY
GUARDIAN PROTECTS
ELECTRONIC SAFETY
LIFESAVER FOR DATA

$595

SAVE YOUR DATA FROM POWER OUTAGES!

BACKUP FOR YOUR COMPUTER, MONITOR, PRINTER AND 5¼" FLOPPY AND HARD DISC DRIVE

• Automatically stops annoying problems from power line interruptions and brown outs • You need standby power to save data • Maintenance free backup power available in 115 volt or 220 volt • 50 or 60 HZ • 150 watts • Complete versatility — operate your system from a 12 volt source, i.e., automobile cigarette lighter, boat or airplane • Rugged self contained gel cell battery • No voiding warranty — no cutting wires • Automatic audio alarm warning tone during commercial power failure or interrupt • UL listed • FCC approved • Transient voltage suppressor gives added insurance from line voltage spikes, utilizing Zener Ray™ • Green/red LED power status indicator • Green — normal AC line power • Slow blinking red — at least 6 minutes of remaining standby power • Fast blinking red — approximately 2 minutes of remaining battery power • Solid state technology unexcelled by any UPS power unit in its class.

RH ELECTRONICS, INC.

COPYRIGHT © 1981 - PATENTS PENDING

566 IRELAN, BUELLTON, CA 93427

(805) 688-2047

SEE YOUR RH ELECTRONICS PRODUCTS DEALER

FOR YOUR APPLE II*:

SUPER FAN II™	$ 74.95
SUPER FAN II™/ZENER RAY™	$109.00
SUPER RAM II™	$125.00
RH 12 VOLT TRANSVERTER	$149.00

FOR MICRO COMPUTERS:

GUARDIAN ANGEL™	$595.00

A Call -A.P.P.L.E. TUTORIAL:

BASIC Suggestions

Luis M. Hernandez

OUT here in the Phillippines we do not grow apples, but do have Apple] ['s and we too are getting caught up with the *Apple-mania* which now plagues many nations. BASIC is one of the languages spoken by micro-computers and because of it, people like you and I who deal with these wonderful machines can speak in this universal language which is so precise that it transcends all races, nationalities and physical barriers.

Using BASIC, we are able to communicate ideas, procedures, systems or whatever, without going through the usual problems of semantics, context or "hidden" meanings behind words. We can write the language in a straight-forward manner and a "statement" made in BASIC can only take on a single meaning, whether it be a Filipino, an American or any other who may happen to read it.

Rosy as it may seem, there are still problems - not in the language itself, but in learning and becoming proficient in it. Therefore, I wish to pass on my experiences as a beginner who tumbled and fumbled through the basics of BASIC. I dedicate this article to those novices who have the willingness and determination to learn. I offer several suggestions which I have learned the hard way and hope that those who are just starting will avoid the long route and achieve proficiency in the quickest possible time.

I. Suggestions On Over-All Program Structure And/Or Techniques

1. Always begin your program with a TEXT and a HOME. The HOME offers a "clean slate" so to speak before any output is written on the video screen. The TEXT command insures that the screen is a full 24 x 40 characters. Without it, your program output may act funny since a previous program may have altered "window" margins on your screen.

2. If you wish to center a line on a screen, please do not count the letters and spaces like I did before (my wife actually thought I was going mad when she saw me talking to myself an counting with my fingers like a first grader). To center a printed message, simply use a string variable and let the computer do the rest like so:

```
100  A$ = "YOUR MESSAGE"
110  PRINT SPC((40-LEN(A$))/2);A$
```

Alternatively, you can use

```
120  PRINT TAB ((40-LEN(A$))/2)
     ;A$
```

or,

```
130  HTAB ((40-LEN(A$))/2):
     PRINT A$
```

If you happen to have an 80-column board, simply change 40 to 80. It also works with line printers although line 130 above won't. Stick to the other alternatives when centering print on a line printer. Okay? I hope from now on you stop counting letters when centering messages or titles on the screen

3. If you are developing a program with many variable inputs, you may find that somewhere about mid-program, you key in the wrong value and then pressed "RETURN". Having realized your mistake you now bang the table (not the Apple, I hope) and start wishing you could go back to the previous line and change the input value. You have two choices: a) re-run the program from the start, or b) continue and let the program take its natural course to the end, which undoubtedly will give the wrong results. Do not despair, for I too have experienced this. I devised a so-called "double-return" technique which gives me a chance to confirm an input variable by having to press "RETURN" twice before the program proceeds. Thus,

```
LIST

110   TEXT : HOME
120   VTAB 10: PRINT
       "ENTER VALUE OF
       X=";
130   HTAB 20: INPUT
       20: INPUT "";A$
140   IF A$ = "" AND
       FLAG = 0 THEN
       120
150   IF A$ = "" AND
       FLAG = 1 THEN
       170
160 X =  VAL (A$):FLAG
       = 1: VTAB 10:
       GOTO 130
170   VTAB 10: HTAB 20:
       PRINT X:FLAG =
       0
180   REM  REST OF YOUR
       PROGRAM
```

The above routine allows recovery from:

a. Pressing "RETURN" without keying in a value yet;
b. Keying in a wrong value and pressing "RETURN".

Of course, if you haven't pressed "RETURN" after keying in a value, you can always use the old "left-arrow" key to change the input value. This routine also helps those, like myself, with big fingers who always seem to hit keys other than those intended when typing.

You may also note that we used a "FLAG" variable, which brings us to the next topic. —◉→

4. Flags are very useful in programming because they, together with the IF...THEN statement, can control program flow quite effectively. Flags are very similar to "signals" which tell the computer which program branch to process next. For example, if you were driving your car towards an intersection, you will notice that the traffic light controls your movement as well as those of other cars. A red light tells you to stop, a green light tells you to go straight, a left-arrow to turn left, etc. Program flags work the same way. Look at line 140 above. This line tests if FLAG is 0 and if A$ is null. These conditions are true if no value was keyed in and a "RETURN" was pressed, thus creating an error condition. This line simply recovers from that error and you are again prompted for the value of X in line 130.

   If a value has been keyed in, then A$ is not equal to null. This brings us to line 150 which again tests A$. Since A$ is not null, then line 160 is processed. Here A$ is converted to its numeric value and stored in X. Note also that in the same line, the variable FLAG is set equal to 1 in preparation for the second pass (the input confirmation pass or your second "RETURN"). Since you are transferred back to line 130, you have the option of keying in a value or "confirming" the previous value. If you simply press "RETURN" at this point, A$ is set to null and line 140 is again processed. Since you have set FLAG equal to 1 in line 160, the test for FLAG in this line will yield a false and you end up in line 150. In this line you will note that both conditions are true during this second pass, and program execution is directed to line 170 which prints the value of X, resets FLAG to 0 and goes on with the rest of your program.

   There, that was simple wasn't it? So you see, the BASIC language is beautiful because in so very few words, much meaning is communicated.

5. Having discussed the use of flags, we now can discuss the use of GOSUB's as useful programming techniques. GOSUB's are unique in that they save memory space (you don't have to use those same lines over and over again throughout your program). GOSUB's can be used for common print routines, common input routines, common screen messages, etc. Try to study the short program below:

```
]LIST
10   TEXT
20   HOME
25 I = 1: VTAB 1
30 A$ = "ENTER FIRST
      NUMBER=": GOSUB
      1000
40 A$ = "ENTER SECOND
      NUMBER=": GOSUB
      1000
50   FOR I = 1 TO 2
60   VTAB I: HTAB 21
70   INPUT "";A$
80   IF A$ = "" THEN 100
90 A(I) =  VAL (A$)
100  VTAB I: HTAB 21
110  PRINT A(I)
120  NEXT
130 A(3) = A(1) + A(2)
140  PRINT :A$ = "SUM
      IS=";FLAG = 1:
      GOSUB 1000
145  VTAB 23: CALL - 868
150  VTAB 10
160  PRINT "SELECT NUMBER:
     ":PRINT
170  PRINT "1. EDIT
      INPUT DATA"
180  PRINT "2. EXIT"
190  GET X
200  IF X < 1 OR X >2
      THEN PRINT  CHR$
      (7): GOTO 190
210  ON X GOTO 220,240
220  HOME : VTAB 23:A$
      = "PRESS 'RETURN'
      TO ACCEPT DEFAULT":
      GOSUB 2000
230  GOTO 25
240  END
1000  PRINT  SPC( 20
      -  LEN (A$));A$;:
      IF FLAG = 1 THEN
       PRINT A(I);
1010  PRINT
1020 I = I + 1
1030  RETURN
2000  PRINT  SPC( (40
      -  LEN (A$)) /
      2);A$
2010  RETURN
```

   The above program uses several useful concepts in programming. The first is the use of the GOSUB. The GOSUB's were used in lines 30, 40 and 140. If the steps in lines 1000–1030 were not used as "GOSUB" routines, then you would have had to write lines 1000–1020 three times (i.e., one for each occurrence of the GOSUB in lines 30, 40 and 140). So the use of GOSUB's are indeed very useful in saving memory space, time and effort. There is, however, a trade-off, since GOSUB's require longer to execute than straight line code. GOSUB 1000 and GOSUB 2000 are what you may call "formatting" routines. These are simply routines to determine where your input messages and output will be placed on the screen. Can you guess what GOSUB 2000 does? (No, don't tell me you forgot!).

   Let's try and examine line 30 and 40 above. Instead of providing direct statements like:

```
30  INPUT "ENTER FIRST
    NUMBER = ";A(I)
```

   we decided to go via the "string" approach. Its advantage is that all input variables required for computation are immediately displayed and the user immediately knows what inputs are required of him.The "cursor" (you know, the blinking square on the screen) is positioned by, and serves as a prompt for the next variable to be entered. This is the second concept: the so-called "fill-in-the-blanks" input approach. Had we used the "INPUT" statement directly in lines 30 and 40, the input prompts would have appeared one at a time, namely, the prompt for the "second number" would appear only after the value for the first variable ("first number") had been entered. But then one may say at this point that it really is a matter of choice as to which method of input to use: direct input statements or "fill-in-the-blanks." Before you pronounce judgement, read on and think about the next related concept.

   The "fill-in-the-blanks" input method prepares you for the program's ability to accept changes in input without having to type "RUN" again after the computer completes processing. This approach will enable you to provide your program with "what if..." capability. But before going into this in detail, look at lines 150–200. These lines actually generate a "MENU" which is the third concept applied in the program above. Although it differs from the menu offered by McDonald's, it still offers "courses" which are really

courses of action or alternatives from which to choose. The use of a "menu" concept is helpful in that it simplifies your programming task. Programming is done a lot easier if you break up the whole task into simple, uncomplicated chunks. If you examine the large professional programs currently available on the market, they may look complex at first, but in actuality they are made up of smaller chunks which are simple as can be; in fact, more often than not, they are as simple as the programs you and I write. Yes, even professional programmers use menus. I think it would do you good to follow in their steps and think in terms of small chunks which you can later put together to form more complex structures.

The menu of the program above offers us two choices: to "Edit Input Data", or to "Exit." Let's examine the menu more closely. Lines 150–180 are simple enough. They simply print the courses of action available to you. Line 190 stops program execution and waits for your choice. After entering your choice, line 200 "validates" your input. This simply means it tests whether your entry is acceptable. If not, a bell (beep?) is sounded and program execution is again halted to await your entry in line 190. Once your entry is found acceptable, line 210 directs execution accordingly. If your choice was no.2, then the program terminates in line 240. If the choice was no.1, then the message "PRESS "RETURN" TO ACCEPT DEFAULT" is printed at the bottom of the screen. "DEFAULT"...Hmmmmmm... yes, we often see this word in computer manuals and other articles. Can this be a game where one team fails to show up and loses by "default"? True, losing by "default" means you lose for "failure to act" or "failure to appear" for the scheduled game. In our program, the "default" value of a variable is that value which is currently assigned to it. So if we simply press "RETURN", then the variable's value is unchanged because we "failed to act" or change the current value. "Failure to act" = "Default"="Unchanged"—see, very simple, isn't it?

Getting back to line 30's GOSUB 1000 statement, you will notice that because the variable FLAG has been set equal to 1 in line 140, the existing value of variable A(I) will be printed after its corresponding input prompt on the screen. These are the "default" values. When the program halts for input in line 70, the cursor will be blinking over the first digit of the "default" value. If you wish to change this value, key in a new number and press "RETURN"; otherwise, simply press "RETURN" with no value keyed in and you will see the "default" value remain unchanged as the cursor advances downward to prompt for the next variable input. This whole process occurs in lines 70-90 where A$ is tested for a null (a condition where a "RETURN" was pressed with no new value keyed in), in which case execution is directed to line 100 where the previous or existing value of A(I) is printed. If a new value had been keyed in, the test for A$ as null would be false and the new numeric value of A$ is assigned to A(I), thus eliminating the old value. "Default" is the fourth concept we have applied.

The combined use of the "fill-in-the-blanks," "menu" and "default" approach can make your programs quite flexible and easier to use because you are able to recover from input errors and change input values with fewer keystrokes and answer "what if..." questions without having to re-type "RUN" or input all variables again from the start. Its usefulness becomes greater as the number of your input variables increase; not to mention the "professional" impression your programs give. If you are a novice at programming, spend some time studying the program above. It's not too long but the knowledge gained will carry you far. Study its GOSUB's, FLAG's, screen format, "default" and "menu".

II. Suggestions On Print Routines

1. As novices, we have all gone through the discovery of how the Apple text screen works. We have learned the effects of commas, semi-colons and the "carriage return" in PRINT statements and how these affect screen format. You probably know that when printing at line 24 of the screen, the next carriage return encountered will scroll up to the contents of the screen thereby losing the printed message in line one (if any).

We have also labored to perfect our screen format only to find out that when we have more than 24 lines to display, the program continues to scroll output up until the last PRINT is executed; but, alas, we have lost the top portion of output at a speed too fast for our eyes to catch, much less for our brains to comprehend. Consider the program below:

```
]LIST

10  TEXT
20 A = 1:B = 0:C = 100
30  IF C - B <  = 23
      THEN B = C: GOTO 50
40 B = B + 23
50  HOME
55  FOR I = A TO B
60  PRINT I, INT (I
      ^ 2), INT (I ^
      3)
70  NEXT
80  PRINT "PRESS ANY
      KEY TO CONTINUE";:
      GET A$
90  IF B <  > C THEN
      A = I: GOTO 30
100  END
```

An alternate program would be:

```
]LIST

10  TEXT
20 C = 100
30  FOR I = 1 TO C STEP 23
40  HOME
50  FOR J = I TO I +
      22
60  IF J > C THEN 80
70  PRINT J, INT (J
      ^ 2), INT (J ^ 3)
80  NEXT
90  PRINT "PRESS ANY
      KEY TO CONTINUE";:
      GET A$
100  NEXT
```

The first program above is designed to print out a table of squares and cubes for numbers 1 to 100. Line 30 tests if the remaining number of lines to be displayed is greater than or equal to 23 (the number of lines allocated for screen print out). If so, the upper limit of the loop "B" is increased by 23. After 23 lines have been printed, it prompts with a message in line 80. Please note the very important application of the semi-colon (;) after the PRINT statement in this line. This prevents a carriage return from being sent to the screen and scrolling up the text. Line 90 tests if all of "C" lines have been displayed. If not, the lower limit "A" is likewise increased and execution proceeds in this manner until all "C" lines have been printed in batches of 23 lines per batch.

BASIC SUGGESTIONS

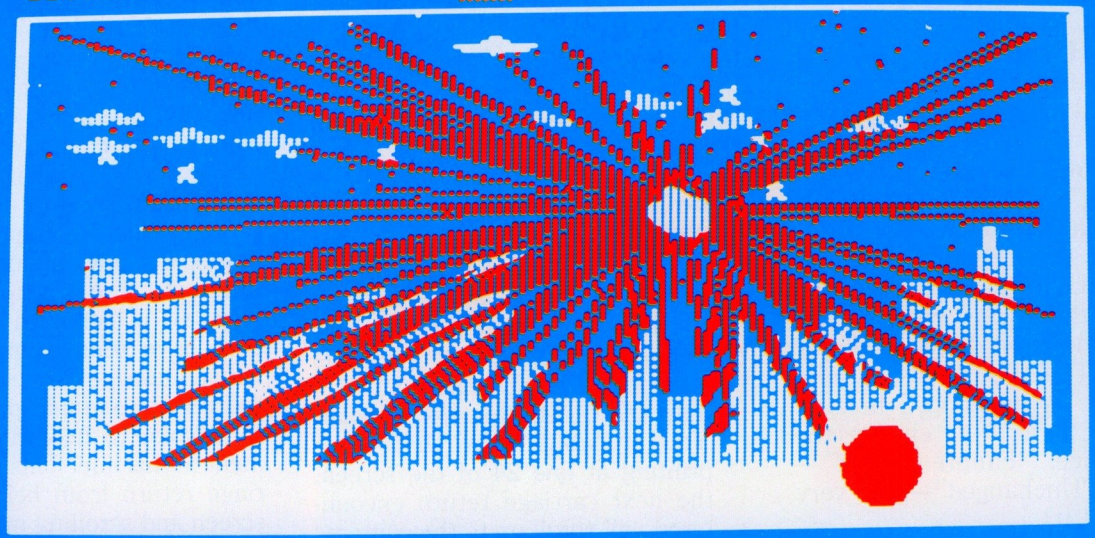
2. Like many novice BASIC programmers, after having perfected their video print routines, separate print routines are written to transfer the screen contents on to a line printer. Thanks to an article written by Bill Parker, "Examining the Video Text Buffer," *Call -A.P.P.L.E.*, February, 1982, a BASIC routine to dump the screen on to a line printer has been inspired in me. Much of the work has been done by Mr. Parker in mapping the Apple text screen. The same issue of *Call -A.P.P.L.E.* also contains an article on a "Text Screen Dump" by David W. Walker, but it is written in machine language. If one does not mind a slight delay (7 seconds) before the line printer starts to work, then the BASIC routine may be his logical choice to dump the screen contents on to a line printer. For the sophisticated, machine language lovers, you have Mr. Walker to thank for the excellent job he has done in providing us a text screen dump routine. (**Note:** *For the program below, please change line 90 accordingly to accommodate your printer slot number).*

Novices, let's move ahead now and please, if you have your own techniques and suggestions, pass them on to even newer beginners (and to us, too, for we can also use them). It is the only way to go in this brotherhood of Apple lovers.

```
]LIST

10   DIM A(960):T = 1
20   FOR K = 1024 TO
       1104 STEP 40
30   FOR I = K TO K +
       896 STEP 128
40   FOR J = I TO I + 39
50 A(T) =  PEEK (J):T
       = T + 1
60   NEXT J
70   NEXT I
80   NEXT K
90   PRINT  CHR$ (13);
       CHR$ (4);"PR#1":
       PRINT  CHR$ (0);
100  FOR I = 1 TO 960
       STEP 40
110  FOR J = I TO I
       + 39
120  PRINT  CHR$ (A(J));
130  NEXT J
140  PRINT
150  NEXT I
160  PRINT  CHR$ (13);
       CHR$ (4);"PR#0"
```

3. A last note for novices: Take your precious disks out of the drive(s) and replace them with "scratch" disks or any disk you don't mind "wiping out" data from. *Then,* monkey around with the computer; experiment with your programs; don't be afraid to try anything. As many people have said, there is nothing you can do at the keyboard (short of using a sledge hammer on it) that will damage your precious Apple. Much of our knowledge today stems from accidental discoveries made by people who are moved by gut feelings to POKE (or PEEK?) into the unknown. If you run out of ideas, read books, articles or whatever you can lay your hands on. Better yet, join *Call -A.P.P.L.E.* It is made up of people who like to POKE and love to share their knowledge.

a city dies whenever nightfalls



As Commander of the Emerald Cityscape,

How many nights can you survive?

Are you really good enough for this one? Can you make it through to just one more dawn? Nine levels of play from beginner to impossible. For the Apple II or II+, 48K with paddles or joystick.

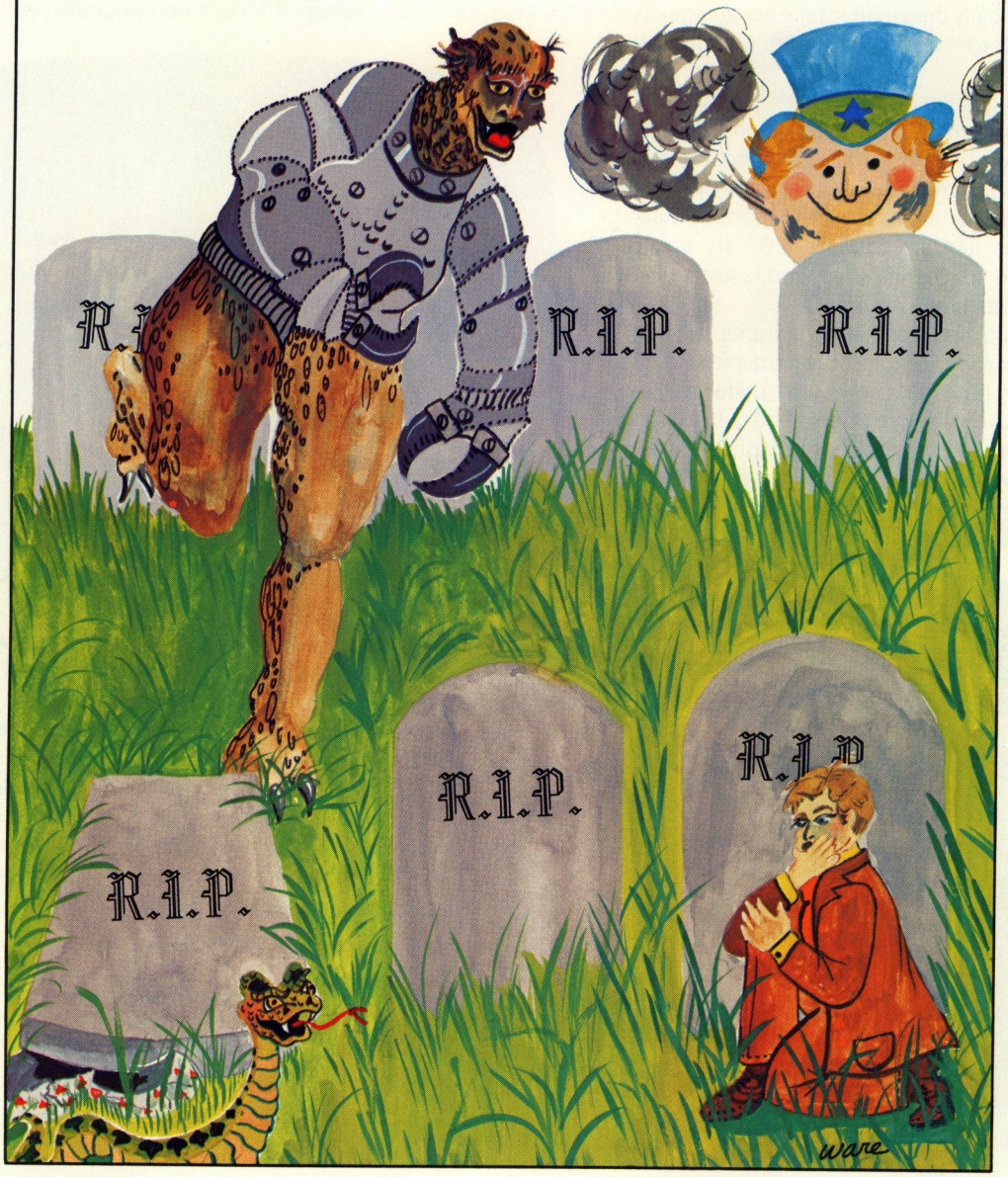
See NIGHTFALLS at your dealer or order direct for $29.95. Mastercard and Visa holders order toll-free, 1-800-835-2246.

OMEGA MICROWARE, INC.™

222 So. Riverside Plaza
Chicago, IL 60606
312-648-4844

Apple II and Apple II+ are registered trademarks of Apple Computer, Inc. NIGHTFALLS is a trademark of Omega MicroWare, Inc.

MONSTER MASH



The Software Farm
3901 So. Elkhart
Aurora, CO 80014
PH: (303) 690-7559

It is late at night in a monster infested graveyard and you have been given the job of keeping the monsters in. All you have between you and complete chaos is a new Monster Masher System and quick reflexes.

Moster Mash is an original and unique arcade action game written in assembly language for the Apple II and Apple /// (in emulation mode).

$29.95

DATA CAPTURE 4.0©

The most advanced and easiest to use telecommunications program for use with the MICROMODEM II™ or the Apple COMMUNICATIONS CARD™

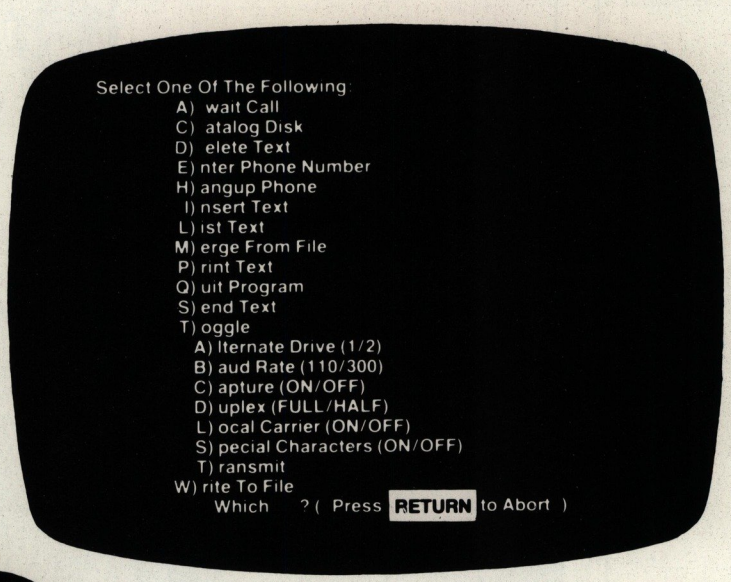
If you use your Apple II™ computer to communicate with other Apples or with timesharing systems, then you need Data Capture 4.0.

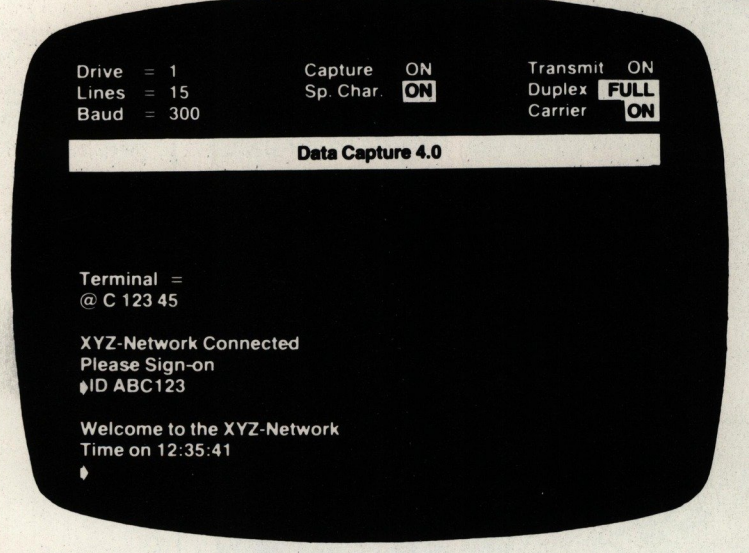
If you want to save stock quotes, airline schedules, electronic mail, or other data from a remote computer system for future use, then you need a smart terminal program for your Apple II™. Data Capture 4.0 is the most user friendly and most flexible program of this type. It is specifically designed to take advantage of all the features of the Micromodem II™, but it works equally well with the Apple Communications Card™ and several other popular serial interface cards.

Data Capture 4.0 is the only Apple II™ smart terminal program available which is fully copyable and modifiable. This means that you don't have to worry about backup. Go ahead and make all the backup copies you need.

The unprotected format and helpful documentation make modifications to Data Capture 4.0 very easy to perform. In addition, full technical support is available by phone from Southeastern Software.

Data Capture 4.0 has many other features. Incoming data files are automatically captured regardless of length. Data in the memory buffer can be viewed, edited, printed, saved to or loaded from disk, or transmitted to the remote system at any time. An unattended mode of operation is provided so that you can call your Apple from another location and send data to it or load data from it. Data Capture 4.0 is fully compatible with the Apple III™ in emulation mode. It is also compatible with all popular lower case adapters for the Apple II™, including the widely used shift key modification. An automatic logon utility for use with the SOURCE is provided with Data Capture 4.0. Also included is a HELP text file containing the latest tips and suggestions on using Data Capture 4.0 effectively (many of these are the result of feedback from customers).





See your local Apple dealer today for a demonstration of Data Capture 4.0 or order direct from Southeastern Software at the address below.

(Please include $2.50 for postage and handling.)

If you presently own a previous version of Data Capture you may upgrade to the new version for the difference in price plus $2.50 for postage and handling. MASTER CARD or VISA orders may be placed by phone.

DATA CAPTURE 4.0 ..... $65.00
DATA CAPTURE 4.0 / 80* $90.00

*Specify either Videx, Smarterm, Double Vision or Sup 'R' Terminal version

— Dealer Inquiries Invited —

Requires DISK II™, Applesoft II™ and 48K of Memory

DATA CAPTURE 4.0© 1980-Southeastern Software

*Apple™, Apple II Plus™, Disk II™ and APPLESOFT II™ are trademarks of Apple Computer Company.
SOURCE – Source Telecomputing Corporation.
*Micromodem™ is a trademark of D.C. Hayes Associates, Inc.

Southeastern Software
6414 Derbyshire Drive ● New Orleans, Louisiana 70126
Dept. CA (504) 246-8438 or (504) 246-7937

master charge THE INTERBANK CARD
Bank Americard VISA

A Call -A.P.P.L.E. TUTORIAL:

Doing It With A Naked Apple

RELOCATING MACHINE LANGUAGE

Bill Starbuck

MY 16K card was doing me almost no good at all, so I was elated to see Cornelis Bongers' DOS MOVER in the August 1981 issue of *Call -A.P.P.L.E.* DOS MOVER transfers the DOS program into a 16K card or a language card, fills the remaining space on the 16K card with five input-output buffers, and places transfer instructions at the top of the main memory. Bongers also explained how to modify the Program Line Editor (PLE) so that it can be loaded into the space just below the DOS transfer instructions.

However, these changes left me no better off than before because I use the Program Global Editor (PGE) as well as PLE. Since PGE continued to load into the same locations as before, HIMEM remained the same, and 10.5K of space lay inaccessible between PGE and PLE. The center column in Figure 1 shows the problem.

To solve this problem, I relocated PGE so that it snuggles up against PLE. The experience presented enough challenges to keep me interested, it taught me a lot about machine language, and it gave my Apple 10K more space for Applesoft programs.

You would probably enjoy and learn from a similar relocation project; this article tells you how to go about it. I have phrased the basic guidelines in general terms so that you can apply them to most machine-language programs. PGE, however, affords a concrete example.

You do not have to know much about machine language, either to read this article or to start relocating a program. But you should have read an introduction (such as Don and Kurt Inman's *Apple Machine Language* or Robert Mottola's *Assembly Language Programming for the Apple][)* as well as chapter 3 in your *Apple][Reference Manual.*

Relocating machine language becomes substantially easier and more elegant when you use a disassembler (such as Sourceror) and an assembler (such as Big Mac). But you do not need such programs, you may not own such programs, and you will learn more about machine language without their help. Therefore, this article explains how to relocate a program using only the Monitor in the Apple][+. A subsequent article will tell you how to use Sourceror and Big Mac to facilitate relocation and to improve a relocated program.

Step 1. Investigate the co-resident programs.

You can find challenge enough in modifying one program at a time, and too much challenge in modifying co-resident programs simultaneously. To keep the challenges achievable, you should modify co-resident programs sequentially. Consequently, before choosing a new location for a program, you should find out what memory locations its co-resident programs use.

This investigation offers you opportunities to learn more about how the Apple uses memory space. Acquaint yourself with William Luebbert's book *What's Where in the Apple?* with pages 6-8 and 127-131 of A.P.P.L.E.'s book *All About Applesoft,* and with chapter 8 of Don Worth and Pieter Lechner's book *Beneath Apple DOS.* Although the manuals for programs rarely give memory-usage information, why not look? You may find it useful to draw a map summarizing what you find out.

Since I wanted PGE to butt up against PLE, I had to make sure of PLE's lower limit. The PLE manual includes a memory map, but a rather vague one that left me with several questions. I could expect PLE to use memory locations below $800, for all programs make some use of the first seven pages of memory ($0000-07FF). Moreover, Bongers' relocated PLE set HIMEM = $B900. Since the Applesoft interpreter uses the memory just below HIMEM for string storage, I inferred that the relocated PLE made no use of the space just below $B900. Although my inference happened to be right, my reasoning overlooked the fact that using PLE destroys the values of all Applesoft variables. Do you see the flaw in my reasoning?

Step 2. Find out how the program loads.

A machine-language program often gets loaded by an Applesoft program that you must modify. Although the time to modify the loading program comes later, you ought to examine this program at the outset. In particular, you had better interpret any PEEKs or POKEs.

For example, to load PGE, you run a short Applesoft program named PLE.EDIT.LOAD. This program (a) creates a screen display, (b) BLOADs a machine-language program named PLE.EDIT.PROD, and (c) POKEs several numbers into memory. One group of POKEs puts:

76	**into address 1013**
00	**into address 1014**
135	**into address 1015**

To make sense of these numbers, you have to translate them into hex:

3F5- 4C 00 87 JMP $8700

What's Where in the Apple? or David Lingwood's list on pages 127-131 of *All About Applesoft* will tell you that $3F5-3F7 contains the instruction that is executed when someone enters an ampersand (&). PGE can be activated by entering &, and these POKEs tell the Apple to start executing PGE at location $8700.

PLE.EDIT.LOAD also pokes locations 111, 112, 115, and 116. A little more research discloses that the program is setting HIMEM = $73F0.

Step 3. Load the program and find it.

Load the machine-language program by entering BLOAD PROGRAM. NAME, then enter CALL-151. The asterisk (*) prompt indicates that you are interacting with the Monitor. If you are working with the unrelocated DOS, enter AA60.AA73: the bytes in $AA60-AA61 specify the length of the program you just loaded, and the bytes in $AA72-AA73 specify the address where the program starts. For example, after BLOAD PLE. EDIT.PROD, I found:

```
AA60- 00
AA61- 1C
AA72- 00
AA73- 74
```

These say that the program begins at $7400 and has a length of $1C00, implying that it ends at $8FFF.

Can you verify that ending address? The Monitor will help you make calculations in hex: if you respond to the Monitor prompt by entering a one-byte addition or subtraction problem, the Monitor will tell you the answer. For example,

```
*74+29
=9D
```

But try two bytes: 7400+1C00 or 8FFF-1C00!

If you are working with a relocated DOS, finding your program can be fun. In the December 1981 *Call -A.P.P.L.E.,* Cornelis Bongers said that, after using DOS MOVER, one could find the starting address in $BBFE-BBFF and the length in $BBFC-BBFD. Acting on this information, I ran DOS MOVER, loaded PLE.EDIT.PROD, and looked at the specified addresses. I found:

```
BBFC- D6
BBFD- 60
BBFE- C9
BBFF- 9A
```

These said the program started at $9AC9, was $60D6 bytes long, and extended to $FB9E. These would have been incredible figures even if I had not known what the figures should be: do you see why they have to be wrong? My first conjecture turned out to be right: all of the other DOS MOVER addresses lay between $BF00 and $BFFF, so perhaps BBFC should be BFFC. There they were:

```
BFFC- 00
BFFD- 1C
BFFE- 00
BFFF- 74
```

Some relocated DOSs use other addresses. The Memory Management System sold by Computer Data Services, for instance, puts the DOS transfer instructions in page 3 ($300-3FF); so you ought to look in page 3 for the start and length data. My experience suggests that you would be wise to start by obtaining the correct data from an unmodified DOS, and then to verify these data with the relocated DOS.

Step 4. Search the program for references to addresses outside its own range. (Pass 1 through the program)

You need not study a program before starting to move it. I, for one, lack the patience to do this; and you can relocate a program without digging into many of its details. You will have to comb through the program several times during the relocation process; and if you keep your eyes open, you will learn more about the program during each pass. After several passes, you will discover that you understand a great deal.

Start by searching the program for the answers to a key question: what memory locations does the program use outside its own range? You will recall, for instance, that the original PGE starts in $7400 and extends to $8FFF. But PGE might well use memory locations below $7400 or above $8FFF. In fact, it does both.

After you complete step 3, you have the program in memory, you know where it starts and stops, and you are looking at the Monitor prompt (*). You can list the program by entering its starting address followed by L; for PGE, this would be 7400L. So make that entry, and the computer will disassemble and display the twenty instructions shown in listing 1.

For the time being, you can ignore all two-byte instructions. Focus your attention on the center column which contains just a few numbers: the high-order bytes of addresses in three-byte instructions. The addresses in three-byte instructions fall into three categories:

1. One category is references within the program's range — addresses within $7400-8FFF in the case of PGE. The JMP from $740B to $876A is an example. You can ignore these until the next time you go through the program.
2. You can also ignore for the present instructions that refer to the Monitor, the Applesoft interpreter, DOS, or special addresses. For instance, the instructions at $7403 and $7420 both refer to the Monitor, the instruction at $7408 refers to the keyboard-input buffer, and the instruction at $740E refers to the keyboard strobe. *What's Where in the Apple?* identifies such addresses.

Listing 1

```
7400- 8D E0 92   STA  $92E0
7403- 20 58 FC   JSR  $FC58
7406- A9 CC      LDA  #$CC
7408- 8D 00 02   STA  $0200
740B- 4C 6A 87   JMP  $876A
740E- 2C 10 C0   BIT  $C010
7411- A9 07      LDA  #$07
7413- 85 23      STA  $23
7415- A9 00      LDA  #$00
7417- 8D E0 92   STA  $92E0
741A- 85 25      STA  $25
741C- 85 24      STA  $24
741E- 85 08      STA  $08
7420- 20 C1 FB   JSR  $FBC1
7423- A4 24      LDY  $24
7425- B1 28      LDA  ($28),Y
7427- 85 00      STA  $00
7429- 85 07      STA  $07
742B- 29 3F      AND  #$3F
742D- 09 40      ORA  #$40
```

3. Make a list of the residual category, adding some notes about how the addresses are used. In the above listing, only one address falls into this residual: the instruction at $7400 stores something in $92E0, and $7417 stores a zero in the same location.

To continue to the next group of twenty instructions, just enter L: the Apple will supply the starting address.

Every so often, your Apple will interrupt a listing with three question marks. The triple question marks indicate that the corresponding bytes cannot be interpreted as instructions. You should figure out what these sections of code denote. Here is one example:

Listing 2

```
*
7FE6-  A5 6A     LDA   $6A
7FE8-  85 07     STA   $07
7FEA-  60        RTS
7FEB-  A9 7F     LDA   #$7F
7FED-  EA        NOP
7FEE-  87        ???
7FEF-  4E A5 07  LSR   $07A5
7FF2-  C5 0B     CMP   $0B
7FF4-  90 29     BCC   $801F
```

Notice the RTS in $7FEA: evidently something different starts at $7FEB. What might be the meaning of A9 7F EA 87 4E A5 and so on? And where do the instructions properly resume? In this instance, $7FA9 and $87EA are two addresses, the addresses of instructions. The instructions resume at $7FF0; the programmers apparently preferred to start the next subroutine at an address ending with zero, so the $4E in $7FEF is a meaningless, and confusing space filler.

Because addresses like those in $7FEB–7FEE do not stand out in that prominent center column, I call them imbedded addresses. You should keep a list of the imbedded addresses for later use.

RTS is not the only instruction that ought to tell you to look for the start of something: what other instructions do this?

Here is another example:

Listing 3

```
7C5A-  D0 D2     BNE   $7C2E
7C5C-  CF        ???
7C5D-  C7        ???
7C5E-  D2        ???
7C5F-  C1 CD     CMP   ($CD,X)
7C61-  A0 D4     LDY   #$D4
7C63-  CF        ???
7C64-  CF        ???
7C65-  A0 CC     LDY   #$CC
7C67-  C1 D2     CMP   ($D2,X)
7C69-  C7        ???
7C6A-  C5 A0     CMP   $A0
```

When you encounter series of bytes starting with B, C, or D, you are usually looking at numbers or letters. Can you interpret the above example? See page 8 of the Apple][Reference Manual. (If so much nakedness discomfits you, don the small fig leaf that is presented in the appendix to this article.)

Finally, here is a difficult mystery. PGE includes several sections of this type:

Listing 4

```
8EC6-  4C 00 74  JMP   $7400
8EC9-  30 32     BMI   $8EFD
8ECB-  30 31     BMI   $8EFE
8ECD-  00        BRK
8ECE-  0D B8 15  ORA   $15B8
8ED1-  86 43     STX   $43
8ED3-  4D 50 81  EOR   $8150
8ED6-  23        ???
8ED7-  24 30     BIT   $30
8ED9-  30 00     BMI   $8EDB
8EDB-  0D BD 15  ORA   $15BD
8EDE-  86 42     STX   $42
8EE0-  45 51     EOR   $51
8EE2-  81 4C     STA   ($4C,X)
8EE4-  43        ???
8EE5-  58        CLI
8EE6-  58        CLI
8EE7-  00        BRK
```

I spent hours fruitlessly searching the program for nonexistent references to these addresses. If you like mysteries, tackle this one. But do not think that you ought to be able to solve it. I consulted three professional programmers, none of whom offered an interpretation; and not until after I had successfully relocated PGE did I finally figure out what these sections of the program are. (That is a hint.)

If you own a printer, you may want to print a program listing on which you can scribble notes. You can print the listing while you are making this first pass through the program. Enter Ctrl-C to exit the Monitor, turn on the printer, enter PR#1 to direct output to the printer, and then enter CALL-151 to reenter the Monitor. To stop the printing, enter Ctrl-C, PR#0, CALL-151.

Step 5. Decide how much to move the program.

My search of PGE left me with the following information: First, several instructions said:

Listing 5

```
99 00 90     STA   $9000,Y
BD 00 90     LDA   $9000,X
B9 00 90     LDA   $9000,Y
99 00 91     STA   $9100,Y
BD 00 91     LDA   $9100,X
B9 00 91     LDA   $9100,Y
```

The references to $9100 were usually paired with references to $0200, such as

Listing 6

```
B9 00 91     LDA   $9100,Y
9D 00 02     STA   $0200,X
```

or

Listing 7

```
B9 00 02     LDA   $0200,Y
99 00 91     STA   $9100,Y
```

I deduced that $9000-90FF and $9100–91FF were both one-page storage buffers. Do you see how I drew this conclusion? Second, several instructions referred to addresses from $92D0 to $92FF, but none referred to addresses in $9200–92CF or to addresses above $92FF. Third, I found no addresses below $7400. I remembered, however, that the loading program set HIMEM = $73F0, so I decided to play it safe and to assume that something might be happening in $73F0–73FF.

Since the relocated PLE started at $B900, PGE could extend up to $B8FF; and since the highest address used by PGE was $92FF, the simplest relocation would be to translate $92FF into $B8FF, by adding $2600 to all of the addresses. When relocating a long program, you should avoid changing the low-order bytes of addresses: such changes add a lot of calculations, each calculation creates an opportunity for error, and you stand to gain less than one page (256 bytes).

I lacked the good sense to make the simplest relocation. I had hopes of saving the space occupied by the two buffers in $9000–91FF, the empty space in $9200–92CF, and the storage locations in $92D0–92FF. As Butch Greathouse and Garry Reinhardt originally conceived PGE, these locations took up no effective space because they overlapped the DOS input-output buffers. Thus, I decided to put these addresses below the program rather than above it; that way, I could play with these locations later without having to relocate the entire program. Since the original PGE program extended to $8FFF, I would move this upper boundary right up against PLE. $8FFF would become $B8FF, and each memory location would be shifted by $2900.

At this point, you should make a table of before-and-after addresses, such as this one:

—◉→

DOING IT WITH A NAKED APPLE

74→9D	•	90→9A
75→9E	•	91→9B
76→9F	•	92D0→9CC0
77→A0	•	92D1→9CC1
•	87→B0	•
•	88→B1	•
•	89→B2	92FF→9CEF
	8A→B3	

Step 6. Change the overt addresses. (Pass 2 through the program)

It is time to start changing addresses. Go back to the beginning of the program, list the first twenty instructions, and identify the high-order bytes that occur in the before-and-after table. These include both references within the program range and references outside it, but not references to the Monitor, the Applesoft interpreter, or DOS.

Because people forget to do things when they try to perform several tasks at the same time, you should postpone changing the imbedded addresses until the next pass. During this pass, just concentrate on the addresses which stand out in that center column.

To change a byte, respond to the Monitor prompt by entering the address of the byte, followed by a colon (:), and then the desired byte. For example, one of the instructions in PGE is:

```
740B-    4C 6A 87 JMP $876A
```

The before-and-after table says $87 should become $B0, and $87 is located at address $740D, so I entered:

```
740D:B0
```

You can also change several bytes at a time: enter the desired bytes separated by blanks. For instance, the first instruction in PGE is

```
7400-    8D E0 92 STA $92E0
```

The address table says that $92E0 should become $9CD0, so both the low-order and the high-order bytes have to be changed. Since the low-order byte is located at $7401, I entered:

```
7401:D0 9C
```

After changing several bytes, you should list that block of instructions and verify the changes. It is much quicker and easier to avoid introducing errors than to find them through debugging. So check your work very carefully ... then check it again. You will recover the time this takes several times over when you debug the relocated program.

When you save the modified program, give it a new name so that you do not write over the original program.

Step 7. Change the imbedded addresses. (Pass 3)

You probably identified many of the imbedded addresses during the previous steps. Now is the time to comb the program systematically for them and to change the ones you have found. Besides the triple question marks, look for instructions which store data in $36–39, which is where the Monitor and DOS look for return addresses. Because step 8 also focuses on $36-39, you can save yourself work by marking or noting all of the references to these addresses.

What do you make of the following?

Listing 8

```
8F0C-  A0 00     LDY  #$00
8F0E-  B9 20 8F  LDA  $8F20,Y
8F11-  9D 00 02  STA  $0200,X
8F14-  E8        INX
8F15-  C8        INY
8F16-  C0 0A     CPY  #$0A
8F18-  D0 F4     BNE  $8F0E
8F1A-  20 70 FF  JSR  $FF70
8F1D-  4C 00 87  JMP  $8700
8F20-  A0 CE     LDY  #$CE
8F22-  A0 B8     LDY  #$B8
8F24-  B7        ???
8F25-  B0 B0     BCS  $8ED7
8F27-  C7        ???
8F28-  87        ???
8F29-  A0 20     LDY  #$20
8F2B-  70 FF     BVS  $8F2C
```

Step 8. If you are adapting the program to a relocated DOS, change the references to DOS. (Pass 4)

Moving DOS into a 16K card obviously becomes a stimulus for relocating machine-language programs. If you are responding to this stimulus, you must adapt the program to the relocated DOS. You can omit this step if you are relocating a program which already operates correctly with the DOS you intend to use.

DOS references come in two delightful varieties: direct and indirect. Both are easy to find. The direct references stick out into that center column: you only have to look for addresses in the range $9D00–BFFF. The indirect references go through $36–39, so you examine instructions that store data in these locations.

Sometimes you can merely change the DOS addresses, but sometimes you have to add some instructions. Which of these you should do depends on the specific address in DOS and on the program you use to relocate DOS.

DOS MOVER illustrates the issues; see pages 9–10 of the August 1981 *Call –A.P.P.L.E.* If you run DOS MOVER and then list $BF00-BFFF, you will find numerous JMPs and JSRs that are mixed with references to $C081 and $C083. Instructions that mention $C083 twice activate the 16K card, and Bongers created a subroutine at $BF66 that mentions $C083 twice. Instructions that mention $C081 de-activate the 16K card. Thus, JSR $9E81 should become JSR $BF93:

Listing 9

The Original Instructions

```
BF93— 20 66 BF  JSR $BF66 ;activate card
BF96— 20 81 DE  JSR $DE81 ;execute the
                DOS subroutine starting
                at $DE81 (which was
                orginally $9E81)
BF99— 8D 81 C0  STA $C081 ;deactivate
                card
BF9C—60         RTS
```

Because DOS MOVER provides the instructions I needed, I could modify the following instructions from PGE by changing only four bytes:

Listing 10

The Modified Instructions

```
808C-  A9 BD    LDA #$BD ;change BD to
                9D
808E-  85 36    STA $36
8090-  A9 9E    LDA #$9E ;change 9E to
                BF
8092-  85 37    STA $37
8094-  A9 81    LDA #$81 ;change 81 to
                93
8096-  85 38    STA $38
8098-  A9 9E    LDA #$9E ;change 9E to
                BF
809A-  85 39    STA $39
```

On the other hand, the following instructions required a bit of programming:

Listing 10a

```
80E8-  A9 F2     LDA  #$F2
80EA-  8D 60 9D  STA  $9D60
80ED-  8D 5E 9D  STA  $9D5E
80F0-  A9 D4     LDA  #$D4
80F2-  8D 61 9D  STA  $9D61
80F5-  8D 5F 9D  STA  $9D5F
80F8-  AD 3D 81  LDA  $813D
```

```
80E8-  20 66 BF  JSR  $BF66
80EB-  A9 F2     LDA  #$F2
80ED-  8D 60 DD  STA  $DD60
80F0-  8D 5E DD  STA  $DD5E
80F3-  A9 D4     LDA  #$D4
80F5-  4C 7C AC  JMP  $AC7C
80F8-  AD 3D AA  LDA  $AA3D
```

```
837C- 8D 61 DD  STA   $DD61
837F- 8D 5F DD  STA   $DD5F
8382- 8D 81 C0  STA   $C081
8385- 4C F8 A9  JMP   $A9F8
```

$837C–8388 was unused space. Do you understand why I did what I did?

Step 9. BSAVE the modified program.

First, if you have not already done so, give the modified program a different name from the original program.

Second, BSAVE the modified program. Since the program is still in its original location, specify the original starting address, or specify a slightly modified address if you added some instructions at the lower end. Also increase the length parameter to allow for any instructions you added.

Third, BLOAD the modified program, specifying the desired starting address and length. This shifts the program to its new location.

Fourth, BSAVE the program again, using the new starting address and length. Now the program will go into its new location automatically when you BLOAD it.

Step 10. Modify the loading program.

Load, list, and modify the Applesoft loading program to reflect the new addresses. You will likely have to make three types of changes:

a) the program's name in a BLOAD instruction such as PRINT CHR$(4)"BLOAD PROGRAM.NAME",
b) PEEKs and POKEs, and
c) screen displays such as PRINT"CALL 34567" or PRINT"PROGRAM.NAME".

Step 11. Debug! (Passes 5, 6, ...)

If you expect the modified program to run properly, you are an optimist. You probably omitted changes you ought to have made, and you may have introduced typographic errors. Moreover, because it is virtually impossible to excise all of the errors from a long program, the program with which you started may incorporate invisible errors; and relocating the program may cause these errors to manifest themselves.

How could relocation reveal previously invisible errors? Again, PGE provides an example. The instructions in $8E0F–8E1A read as follows:

Listing 11

```
8E0F-C9 EA     CMP   #$BA
8E11-90 03     BCC   $8E16
8E13-4C 46 89  JMP   $8946
8E16-29 7F     AND   #$7F
8E18-99 00 02  STA   $0200,Y
```

However, in two places ($75EC and $8FCB), PGE states:

```
4C 14 8E        JMP $8E14
```

The programmers intended to say JMP $8E16, but they got one digit wrong, and created a jump to the middle of an instruction. This glitch produced no problems while PGE occupied its original location: when the Apple jumped to $8E14, it found

```
46 89           LSR $89
```

so the Apple proceeded to $89 and shifted the bits right, then went on to the instruction in $8E16, which is where it should have gone in the first place. The glitch merely shifted the bits in $89, and this had no visible effect because the Applesoft interpreter only uses $89 for temporary storage. However, when I shifted PGE upward, the instruction in $8E13 became

```
B713-   4C 46 B2   JMP $B246
```

and when the Apple jumped to $B714, it found:

```
        46 B2      LSR $B2
```

This relocated LSR had very visible effects because $B2 is part of CHRGET, an important subroutine in the Applesoft interpreter. The relocated LSR actually disabled CHRGET: can you explain why? (For an explanation of CHRGET, see Cornelis Bongers' article in the March 1982 *Call –A.P.P.L.E.)*

I enjoyed finding and understanding that glitch, and more generally I enjoy debugging. You too will enjoy debugging if you treat it as an intellectual challenge and develop some skill at it. Debugging strongly resembles an adventure game, a game that Sherlock Holmes would love. You look for clues, you make conjectures about the clues' significance, and you test your conjectures. But debugging frustrates a lot of people because they do not know how to do it well.

This is no place for a treatise, but you can take much of the pain out of debugging by obeying two fundamental rules:

1. Decompose the task.
2. Let the computer do the work.

Break the program into autonomous sections, and debug one section at a time. For example, PGE responds to various one-key commands. The commands *, !, and % direct the Apple to a short section at the very end of the program that makes some simple calculations. So I started debugging with this section, where the difficulties would be minimal. I gave only these three commands until this section worked properly. Then I looked for another command which activates an autonomous section of the program.

A map of the program will help you decide what to debug in what sequence. Examine the instructions following the program's entry point ($8700 in the original PGE, $B000 in the relocated PGE). This initial section usually incorporates JSRs and JMPs that identify the starting points of major subroutines. Then examine the instructions following the entry point of each major subroutine, continuing until you see how that subroutine terminates: does the subroutine interact with others or is it autonomous? After you inspect all of the major subroutines, you should be able to classify the subroutines hierarchically. If you find flow diagrams helpful, draw one.

When you are trying to locate a specific defect, do not search randomly. Since very small defects can produce large effects, you may be searching for a needle in a haystack. Develop a systematic search procedure that makes your Apple locate the defect for you. For example, divide the section you are debugging in half, by inserting a jump, and then run half. If the first half works properly, the defect must lie in the other half. Also use your Apple to generate evidence about the defect's character: vary the inputs to the program and see what happens to the outputs.

Your detective work can benefit from a fundamental assumption:

The computer acts logically.

Poor Sherlock Holmes had to contend with the vague and erratic logic of human beings. You do not, and you can take advantage of this fact. Every response your Apple makes has a precise and specific stimulus: when your Apple takes an undesirable action, it is responding to an undesirable stimulus. So use logic to develop conjectures about the stimulus: ask yourself what kinds of instructions could produce the undesirable action. —

Step 12. Make improvements.

By the time you finish debugging the program, you will know a great deal about it. So use the opportunity to improve the program. Your improvements might make the program work significantly better, and so be pragmatically useful. An equally important outcome is your personal sense of accomplishment. Of the changes I made in PGE, for example, the ones which please me most do not alter the program's actions.

Appendix

Here is FIG.LEAF, a machine-language program that will help you interpret those perplexing sections of programs. The disassembler built into your Apple assumes that every byte in memory ought to be interpreted as an instruction code or address; when this assumption fails, the disassembler burps three question marks. FIG.LEAF makes the opposite assumption: that every byte ought to be interpreted as part of a message in ASCII code. In effect, FIG.LEAF automates the table on page 8 of your Reference Manual. BLOAD FIG.LEAF before you begin Step 4. Then, when you encounter a perplexing section, respond to the Monitor prompt by entering 300G.

FIG.LEAF distorts the contents of memory in two ways. Because flashing characters interfere with your perception of non-flashing characters, FIG.LEAF converts flashing characters to inverse. And because Bell ($87), Backspace ($88), Linefeed ($8A), and Carriage Return ($8D) produce irregular lines and distracting beeps, FIG.LEAF replaces some control codes with inverse exclamation points. These control codes often inhabit messages, so you will find that inverse exclamation points likely go together with any adjacent words.

Insert FIG.LEAF into your machine by entering an address, a colon, and some bytes separated by commas. For example,

```
300:A0 FF C8 B9 86 03 20 F6 FD
C9 BA D0 F5 20 6F FD
```

You can enter up to 85 bytes before pressing Return. After you have entered the entire program, proofread it carefully for typographic errors. When you are sure that the program is correct, BSAVE FIG.LEAF, A$0300, L$00C9. Finally, BRUN FIG.LEAF to see what it does. If you tell it to start at $300, FIG.LEAF will look at itself.

FIG.LEAF can also help you in other circumstances. For example, FIG.LEAF highlights the structure of data that have been dumped from a disk, and it renders Applewriter files intelligible.

ASSEMBLER/TEACHER

This program makes your APPLE* teach you assembler! We assume you know BASIC already, and now you'd like to learn APPLE* assembler. You need three things (two heavies and a friend)...

1. The APPLE II Reference Manual (heavy),
2. A good 6502 assembler manual (another heavy: ask for both at your APPLE* dealer), and
3. ASSEMBLER TEACHER, a friendly program that takes you right through the learning barrier!

ASSEMBLER TEACHER will soon have you eating those manuals for breakfast. And you'll never outgrow ASSEMBLER TEACHER, since it has tools an expert can use even more than a novice. Here's how ASSEMBLER TEACHER works:

- ASSEMBLER TEACHER prowls your APPLE*s memory page by page, drawing a map of what it finds.
- ASSEMBLER TEACHER shows your CPU running in slow motion on your screen, so you can see exactly what the operations are doing to the registers and the stack (a great debugging tool).
- ASSEMBLER TEACHER exposes the mysteries of hexadecimal, two's complement, character and other data representations, instantly translating from any version to all others.
- ASSEMBLER TEACHER gives you the concepts you need to cope with assembler, using simple on-line lessons that grab you by the keyboard and make you play your way to knowledge.
- ASSEMBLER TEACHER is written by John Fairfield, a Ph.D. in Computer Science who regularly teaches assembler. A RAM Mini-Assembler is included.

"Rating: A . . . a great help to anyone interested in learning machine language programming . . . and the price is right." Peelings II (Jan. '82)

Ask for ASSEMBLER TEACHER at your dealer, or order from:
ASSEMBLER TEACHER
Computer Works, Inc.
P.O. Box 1111, Harrisonburg, VA 22801
(703) 434-1120

$44.95 MC VISA DOS 3.3

*Registered trademark of Apple Computer, Inc.

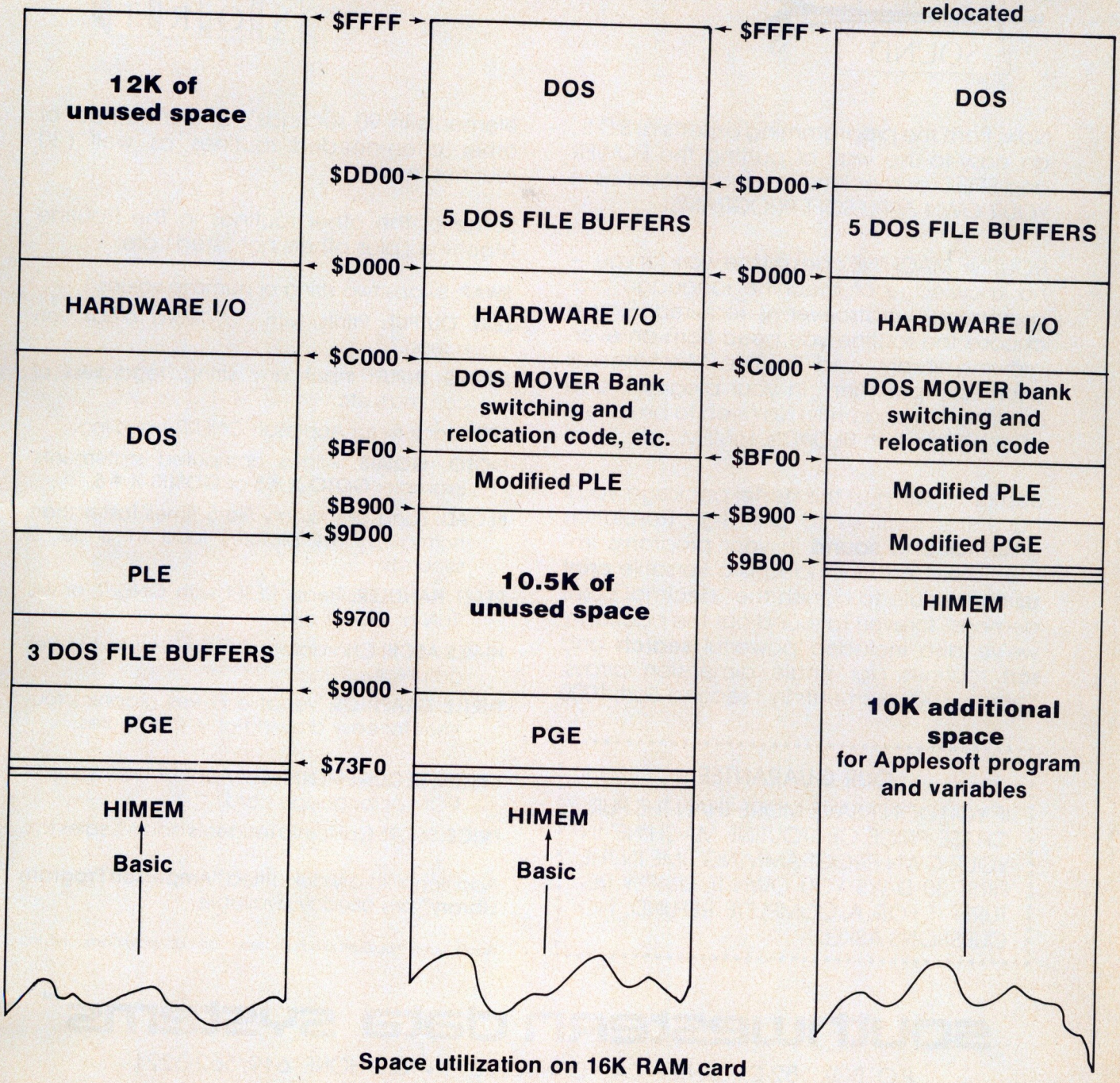
FIG.LEAF hex dump

```
*300.3C8
0300- A0 FF C8 B9 86 03 20 F6
0308- FD C9 BA D0 F5 20 6F FD
0310- 20 C7 FF 20 A7 FF 20 58
0318- FC 20 59 03 20 92 FD A9
0320- A0 20 F6 FD A0 00 20 59
0328- 03 B1 3C C9 8E B0 15 C9
0330- 40 90 11 C9 80 B0 07 E9
0338- 40 18 90 08 F0 C2 C9 87
0340- 90 02 A9 21 20 F6 FD C8
0348- A5 24 D0 DA 18 98 65 3C
0350- 85 3C 90 C5 E6 3D 18 90
0358- C0 A5 25 C9 16 B0 01 60
0360- 68 68 A0 FF C8 B9 A2 03
0368- 20 F6 FD C9 BA D0 F5 20
0370- 0C FD 48 20 58 FC 68 C9
0378- C3 F0 A1 C9 CE F0 BD C9
0380- D1 D0 DD 4C 69 FF 8D C5
0388- CE D4 C5 D2 A0 C6 C9 D2
0390- D3 D4 A0 C1 C4 C4 D2 C5
0398- D3 D3 A0 C9 CE A0 C8 C5
03A0- D8 BA 8D C3 A0 BD A0 C3
03A8- CF CE D4 C9 CE D5 C5 BB
03B0- A0 CE A0 BD A0 CE C5 D7
03B8- A0 D3 D4 C1 D2 D4 BB A0
03C0- D1 A0 BD A0 D1 D5 C9 D4
03C8- BA
```

Figure 1-A
Before DOS Mover

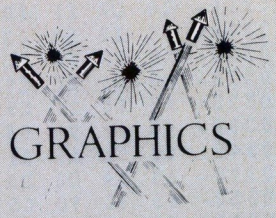
Figure 1-B
After DOS Mover,
with PLE relocated

Figure 1-C
After DOS Mover,
with PLE and PGE
relocated

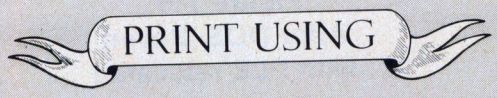


Space utilization on 16K RAM card

THE ROUTINE MACHINE™



A Versatile Programming Utility for the Apple II.



Now, from the programming experts at S.D.S., an easy-to-use way of putting the POWER and SPEED of machine language routines in YOUR OWN APPLESOFT PROGRAMS!

ROUTINE MACHINE does all the work for you — no knowledge of machine language programming, whatsoever, is required. Simply choose the routine you need from an ever-growing library, and Routine Machine will effortlessly put them in your program, and all done transparently! No need to be aware of or bother with BLOAD's, HIMEM:, etc.

Best of all, with just this starter package, you'll have the routines to put High Resolution **graphics** and **sound** in your programs immediately! Also included is a versatile **print using** module to banish the "decimal point demons" forever! To round out the package, we've also included powerful **search** and **sort** routines (for single dimension arrays; Search: 1000 elements in 1 second, Sort: 1000 elements in 90 seconds), and a number of other often-needed routines as well (30 routines in all).

Some of the other routines in The Routine Machine (plus others not listed) are:

SWAP: Swaps two string or numeric values.

TEXT OUTPUT: Prints with no "word break" on screen.

STRING INPUT: Input any string, regardless of commas, etc.

ERR: Stack fix for Applesoft ONERR handling.

GOTO, GOSUB: Allows computed statements. Example: **GOTO X*5** or **GOSUB X*5**.

BLOAD: Load any binary file 5 times faster than normal. Hi-Res pictures load in under 2 seconds.

RESET HANDLER: Treats RESET with ONERR; or will RUN or reboot disk.

HI-RES ASCII: Character set for mixing text Hi-Res graphics.

TURTLE GRAPHICS: Versatile Hi-Res graphics routines for easy drawing of Hi-Res figures.

SYSTEM REQUIREMENTS: 48K Apple II, DOS 3.3 w/FP ROM or language card.

PRICE: $64.95 California residents add 6% sales tax.

Additional library disks titled **"Ampersoft Program Library"** are already available.

OUR GUARANTEE

IF YOU DON'T SAVE MORE THAN THE PURCHASE PRICE OF 'ROUTINE MACHINE' IN YOUR OWN PROGRAMMING TIME IN THE FIRST 30 DAYS YOU OWN IT, SIMPLY RETURN IT FOR A COMPLETE REFUND, NO QUESTIONS ASKED!

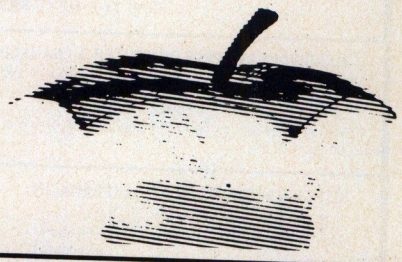
Apple II and Applesoft are both trademarks of Apple Computer, Inc.

southwestern data systems™

P.O. BOX 582-C • SANTEE, CALIFORNIA 92071 • TELEPHONE: 619/562-3221

FROM THE VERY CORE OF APPLE:

The Unknown INTFLG



Kelly W. Puckett

THIS note describes a previously undocumented Applesoft flag and shows how that flag may be used when manipulating Applesoft variables from machine language programs.

The subroutines and flags used here, and their names, are documented in John Crossley's wonderful article, 'Applesoft Internal Entry Points'. *Call -A.P.P.L.E. in Depth 1: All About Applesoft*

There are many occasions when it is desirable or necessary to use an Applesoft variable from a machine language routine. If the variable type — string, integer, real — is known in advance, the method is simple. When the Applesoft textpointer (TXTPTR) is pointing to the name of the variable, the PTRGET subroutine is executed. This subroutine returns with the address of the variable in the A and Y registers and in VARPNT. The machine language program can then do whatever is desired with the value of the variable.

What do you do when you cannot know ahead of time the type of variable PTRGET is going to locate? After executing PTRGET, the Applesoft flag VALTYP ($11) may be tested to determine whether the variable was string or numeric. If it was a string variable, VALTYP will contain $FF. If the variable was a real or an integer variable, VALTYP will contain $00.

But how can you tell whether the variable was an integer type or a real type variable? The existing information on Applesoft's internal subroutines and flags does not offer a single clue to a method.

Clearly Applesoft must be able to tell the difference in order to store and retrieve numeric variables correctly, because the internal representations of the two numeric types are very, very different.

While working on a machine language program which must manipulate all types of Applesoft variables, I discovered another zero page flag set by PTRGET. The flag, which is known as INTFLG, is at $12. If the variable addressed by PTRGET was a real or string variable, INTFLG is left with a value of $00. If the variable was an integer, INTFLG will contain $80.

Testing both variable type flags, VALTYP and INTFLG, will completely determine the type of variable encountered.

The values of the two flags may be added together and saved for later use in a single byte of memory:

Variable	VALTYP	INTFLG	SUM
String	$FF	$00	$FF
Real	$00	$00	$00
Integer	$00	$80	$80

Here is a routine which will locate the variable, and, if the variable is numeric, load the value of the variable into the floating point register, FAC:

Knowing about and using INTFLG will enhance your ability to manipulate numeric Applesoft variables from machine language routines.

```
                         ********************************
                         *                              *
                         * Vartype Finder-Kelly Puckett *
                         *                              *
                         * Call -A.P.P.L.E. : Dec. 1982 *
                         *                              *
                         ********************************
                         *
VALTYP          =        $11
INTFLG          =        $12
VARPNT          =        $83
PTRGET          =        $DFE3
GIVAYF          =        $E2F2
MOVFM           =        $EAF9
                *
                         ORG     $300
                *
0300: 20 E3 DF           JSR     PTRGET          ;Locate variable
0303: 24 11              BIT     VALTYP          ;string or numeric
0305: 10 01              BPL     LOADNUM         ;Branch if numeric
0307: 60                 RTS                     ;Return if string
0308: 24 12     LOADNUM  BIT     INTFLG          ;Real or integer
030A: 30 03              BMI     LOADINT         ;Branch if integer
030C: 4C F9 EA           JMP     MOVFM           ;Value -> FAC
030F: A0 00     LOADINT  LDY     #$00            ;Set index
0311: B1 83              LDA     (VARPNT),Y      ;1st byte of value
0313: AA                 TAX                     ;Save it in X reg
0314: C8                 INY                     ;Index =1
0315: B1 83              LDA     (VARPNT),Y      ;2nd byte of value
0317: A8                 TAY                     ;Put it in Y reg
0318: 8A                 TXA                     ;1st byte in Acc
0319: 4C F2 E2           JMP     GIVAYF          ;A,Y -> FAC

--End assembly--

28 bytes

Errors: 0
```

GET FREE SOFTWARE FOR YOUR COMPUTER!

HOW? JUST ORDER ANY OF THE ITEMS BELOW, AND SELECT YOUR FREE SOFTWARE FROM THE BONUS SOFTWARE SECTION, USING THE FOLLOWING RULE: FOR THE FIRST $100.00 WORTH OF MERCHANDISE ORDERED TAKE 1 ITEM; FOR THE NEXT $200.00 WORTH OF MERCHANDISE ORDERED TAKE ANOTHER ITEM; FOR THE NEXT $300.00 TAKE A THIRD ITEM, ETC. ALL AT NO COST.

HARDWARE by APPLE COMPUTER & FRANKLIN:

APPLE II + 48K	1199	PASCAL DISK + BKS	200
FLOPPY DR. + CNTRLR	545	FLOPPY DRIVE	465
APPLE III 128K	2999	PILOT	125

We carry the rest of the APPLE and FRANKLIN line at low, low prices! CALL!

OTHER HARDWARE for APPLE/FRANKLIN:

D.C. HAYES:

Micromodem II	285	Smartmodem	225
		Smartmodem 1200	CALL

MICROSOFT:

Z80 Softcard	269	16K Ramcard	119

MOUNTAIN COMPUTER:

Expansion Chassis	559	Music System	339
A/D + D/A Card	289	CPS Multi-function	169
X/10 Control Card	169	Super Talker	169

CALL FOR MORE PRICES! WE CARRY FULL LINE!

CALIFORNIA COMPUTER SYSTEMS:

Centronics Par. Int	115	Async. Serial Int.	135

CALL FOR MORE PRICES! WE CARRY FULL LINE!

VIDEX:

80 Col. Bd. & Softswitch	245	Enhancer II	125
Enhancer I	105	Softswitch	29

MORE OTHER HARDWARE FOR APPLE/FRANKLIN:

SSM AIO-II	195	Keybd. Co. Num. Keypad	129
SSM Serial ASIO	115	Sunshine Joystick	39
SSM Par. APIO	99	Game Paddles	29
Novation Apple Cat	319	Shadow / Vet	675
Versawriter Tablet	249	SUP'R MOD	29
Prac. Periph. Microbuff. (32K)	249	Prac. Periph. Microbuff. (16K)	225

OTHER SOFTWARE for APPLE/FRANKLIN:

PERSONAL SOFTWARE/VISICORP:

Visicalc 3.3	195	Visifiles	199

CALL FOR MORE PRICES! WE CARRY FULL LINE!

MICROSOFT:

APPLE Fortran (Z80)	129	APPLE Cobol (Z80)	499
TASC Basic Compiler	139	MBASIC Compiler (Z80)	299
TIME Manager	125	MuMath	199
ALDS	99	M/SORT	149

MICRO-PRO:

Wordstar	225	Mail-Merge	99
Spellstar	149	Data-Star	199
Super-Sort	149	Calc-Star	149

PEACHTREE: BIZ Packages, all ..... CALL

SYSTEMS PLUS: BIZ Packages, all ..... CALL

MORE OTHER SOFTWARE for APPLE/FRANKLIN:

DB Master	179	DB Master for CORVUS	399
ASCII Express	55	PFS	85
Sorcim Super Calc	189	Dakin 5 BIZ Bookkeeper	299
Howard Tax Prep	115	Broderbund Payroll	325
Howard Real Estate Anal	129	BPI Accounting Pkgs/ea	325

WORD-PROCESSORS & SPELLERS for APPLE:

Wordstar CP/M	225	EZWriter Prof. Sys.	149
Magic Wand CP/M	299	MUSE Super Text 80	150
Executive Secretary	199	Wordpower	50
Letter Perfect	125	Hebrew II	55
Magic Window	85	Screenwriter II	110
Spellguard	219	Spellstar CP/M	149

Word Handler (Gives 80-Col. & Lower-Case with no board!) 199

PRINTERS

EPSON:

MX80	499	MX80 F/T	549
MX100 w/Graftrax	729	MX80 Ribbon	15
APPLE Intfce/Cbl	85	GRAFTRAX	60
GRAPPLER Intfc	149	MX100 Ribbon	24

C.ITOH:

F-10 Daisy Wheel (Par)	1495	F-10 Daisy Wheel (Ser)	1495
Pro-Writer (Par/Ser)	599	Pro-Writer (Par)	499
F-10 Tractor Option	225	Printer Interfaces	CALL

NEC:

PC-8023A	495	NEC 7710 Daisy	2345

QUME:

SPRINT 9/45	1995	**DIABLO:** 630 R/O	2099

OKIDATA:

Microline 82A	475	Microline 80	375
Microline Tractor	59	Okigraph I	79
Microline 83A	799	Microline 84 (Par)	1099

IDS:

560 with graphics	1095	Prism-Print Software	49
Prism 80 (Basic)	899	Prism 132 (Basic)	1050
Auto Sheet Feed	125	Prism Color	325
Sprint Mode (200 cps)	125	Dot Plot Graphics	85
SMITH CORONA:	689	**BROTHER HR-1**	CALL

ADD-ON MEMORY CARDS & DISK DRIVES FOR APPLE

MEMORY:

Microsoft 16K Ramcard	139	Saturn 32K Card	199
Legend 128K Ramcard	649	Saturn 64K Card	369
Legend 64K Ramcard	299	Saturn 128K Card	525
SVA 256K APL-Cache	1045	Prometheus 128K	439
AXLON 320K Ram Disk	1149	16K of 4116, 200 NS Mem.	25

APPLE-COMPATIBLE FLOPPIES by MICRO-SCI:

With Controller:		No Controller:	
A35 Exact Replacement	460	A35 Exact Replacement	415
A40 40-Track	489	A40 40-Track	399
A70 70-Track	599	A70 70-Track	499

8" FLOPPY DISK SYSTEMS:

SVA AMS8000 Dual SSDD	1945	SVA AMS8000 Dual DSDD	2595
SVA ZVX4 Quad Cntrllr.	495	SVA Disk 2 + 2 Cntrllr.	359

CORVUS HARD DISKS:

6 MB Hard Disk	2249	Apple Interface	175
11 MB Hard Disk	3945	Other Computer Intfce	CALL
20 MB Hard Disk	4769	Mirror Back-Up	675

MONITORS, PLOTTERS & PERIPHERALS

MONITORS:

Zenith 12" Green	125	Zenith 13" Color	359
Amdek 12" Green (300G)	175	Amdek 13" Color	359
BMC 12" Green	119	BMC 12" Color	349

PLOTTERS:

Watanabe 4-Pen	1150	Watanabe 6-Pen	1400
Strobe Plotter 1-Pen	699	Strobel Apple Intfce	99
Houston Inst. DMP-3	929	Houston Inst. DMP-4	1185

FLOPPY DISKS

Elephant 5.25" Soft SSSD/bx	25	Elephant 8" Soft SSSD/6xr	29
3M 5" Soft SSD/bx	27	3M 8" Soft SSSD/bx	32
Maxwell 5" Soft SSSD/bx	31	Maxwell 8" Soft SSSD/bx	35
DYSAN 5" Soft SSSD/bx	37	DYSAN 8" Soft SSSD/bx	49

COMPUTER SYSTEMS

ATARI

ATARI 800 (16K)	629	PAC-MAN	33
810 Disk Drive	449	ATARI 410	79
16K Ram Memory	89	825 Printer	575
Microsoft Basic	69	850 Interface	169
INTEC 32K Ram Memory	129	ATARI 800 (48K)	739
OSBORNE 1	1695	***KAYCOMP II**	1699

XEROX

Full Line at Rock Bottom Prices. CALL!

NEC

Full Line at Low, Low Prices! CALL!

HEWLETT-PACKARD

Entire Line at Hugh Discounts! CALL!

HARDWARE FOR IBM PC

AMDEK Color II Monitor	769
BABY BLUE Run CP/M!	549
CORVUS Hard Disks	See CORVUS Section this page
CORVUS/IBM PC Intfce	239
MICROSOFT 64K RAM cd	395
MICROSOFT 128K RAM cd	555
MICROSOFT 256K RAM cd	875
QUADRAM 256K, 4-Function Card	699
PERCOM Drives	CALL
TECMAR We carry complete line!	CALL
TANDON 5.25" Floppy Drive SD	249
TANDON 5.25" Floppy Drive DD	329
SEATTLE RAM & RS232 Port	CALL

SOFTWARE FOR IBM

ASHTON-TATE d BASE II	CALL
CONTINENTAL Home Accountant	125
INSOFT Accountant per Module	99
INFO UNLTD Easy Writer II	259
MICROPRO Wordstar	285
SORCIM Supercalc	219
VISICORP Visical C/256K	195
VISICORP Visifile	249
VISICORP Visitrend/Plot	249
VISICORP Desktop Plan	249

GENERAL CP/M SOFTWARE

MICROSOFT:

Basic 80	275	Edit 80	139
Basic Compiler	299	Mu Math/Mu Simp	199
Fortran 80	339	Mu Lisp/Mu Star	159
Cobol 80	499	M-Sort	128
Macro 80	139	Multi-Plan	CALL

MICROPRO:

Wordstar	275	Calcstar	199
Mailmerge	89	Supersort	165
Spellstar	165	Custom. Notes	275
Datastar	239		

ASHTON-TATE:

dBASE II	475	dBASE II Guide	29

SORCIM:

Super Calc	189		

FOX-GELLER:

Quickscreen	129	dUTIL	69

Quickcode (Writes programs for dBASE II) 199

BONUS SOFTWARE SECTION!

SUPER MESSAGE: Creates colorful, dynamic on-screen messages in full page "chunks". Each message allows statements of mixed typestyles, typesizes and colors, in mixed upper and lower case. Styles range from regular APPLE characters, up to double-size, double-width characters with heavy, bold font. Six colors may be used for each different typestyle. Vertical and horizontal centering are available, and word-wrap is automatic. Users can chain pages together to make multi-page messages. Pages can be advanced manually or automatically. Multi-page messages can be stored to disc or recalled instantly $50

APPLE DATA GRAPH 2.1: Plots up to 3 super imposed curves on the Hi-res Screen both X & Y axes dimensioned. Each curve consists of up to 120 pieces of data. Graphs can be stored to disc and recalled immediately for updating. Great for Stock-market Charting, Business Management, and Classroom instruction! $35

APPLE RECORD MANAGER: Allows records within any file to contain up to 20 fields, with user-defined headings. Information can be string or numeric. Users can browse thru files using page-forward, page-backward or random-search commands. Records can easily be searched, altered or sorted at will. Works with 1 or 2 drives. Records or Files can be printed, if desired $40

APPLE LITERATURE INDEX: Allows rapid retrieval (via keywords) of references from total APPLE literature thru June 1981, on 5.25" disc. Each entry in the Index consists of the article, author-name, periodical-name, date of issue, & page nos. The Index is intended to support large magazine files which would require lengthy manual searching to recover information. Annual updates will be available $60

WORDPOWER: Is a simple, powerful, low cost, line-oriented word processor program. It offers a fast machine language FIND & REPLACE. Text can be listed to screen or printer. Lower-case adaptors are supported. You can merge files, move group of lines, and easily add, change, or delete lines. Good for business or personal use. $50

LABELMAKER: Allows a given label to be generated in any quantity from 1 to 32767. The program generates labels not only for addresses, but also for price-tags, part numbers, and mail-messages such as "RUSH", "FRAGILE", etc. A self-incrementing feature allows theatre-tickets to be produced, with a date, and numbers running from a000 to z999. An editor is provided for editing labels prior to printing. All labels may be saved to disc for instant recall $35

APPLE INVOICER: Allows entry of price, quantity, & description of items. Automatically calculates horizontal & vertical totals, figures in sales tax, & special handling. Then it generates required number of invoice copies. Good for people who need arithmetic done automatically! $50

MAIL SUPREMACY: A quick 'n' easy mail list package. Allows a fifth line for foreign country or customer code. Searches & sorts rapidly. Works with 1 drive. $30

APPLE FUNNYWORD: A party amusement! Creates endless artificial comical words on screen or printer! Examples: IDAHEN! UPOTIJI! VUTOHEG! BEJATAV! OKYWEXU! GOPAGA! Good fun for feeble-minded! $20

Above software for APPLE DOS 3.3 only. Call for BONUSES for other systems.

TO ORDER: Use phone or mail. We accept VISA, MC, COD's, personal checks & money orders. Add 4% for credit card. Customer pays handling on COD orders. Foreign orders must be in U.S. Dollars & include 10% for handling. Conn. residents add 7.5% sales tax.

*Not available thru mail-order.

CONN. INFO. SYSTEMS CO.

(203) 579-0472

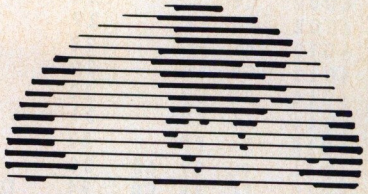
218 Huntington Road, Bridgeport, CT 06608

Not responsible for typographical errors. Prices subject to change without notice.

EXPLORING THE WORLD OF APPLE:

New Uses For the RAM Card

Mark Harris



WITH the cost of memory plummetting all the time, 16K "language cards" for the Apple have become increasingly affordable (I have seen one model priced at $50). Can you make use of the extra memory? One obvious use for the card is for loading languages: for example, the System Master diskette will automatically load Integer Basic into the card when booted. Programs and articles have appeared which allow DOS to be loaded into the language card, freeing the space otherwise occupied in primary memory. But the extra 16K is hard for a beginner to get to: the added memory cannot be addressed (PEEKed, POKEd, etc.) directly without "bank-switching." The purpose of this article is to demonstrate some additional uses of the language card.

The Apple] [is built around the 6502 microprocessor. This chip has 16 address lines, and hence directly-addressable memory is limited to 2↑16 = 65536 bytes (64K). Additional memory can be accessed through a technique called "bank-switching"; by flipping a (soft) switch, the user can "look at" a different bank of memory, but never more than 64K at one time. The 16K card replaces the top $3000 (12288 decimal) bytes of ROM (Applesoft and the Monitor on most systems) with $4000 (16384) bytes of RAM (only $3000 of which is directly available without further switching). The important thing to keep in mind when programming for such a card is that Applesoft and Monitor routines should never be called when the usual ROM has been switched out – havoc would result from such calls. This limits the user (temporarily) to 6502 machine language. One good use of the extra space is for moving a large block of memory to or from the card, since a very short machine language program can perform the move and then return to a controlling BASIC program. The "LC PAGE 2 BACKUP" routine and its BASIC driver "PICTURE MOVE" listed below give an example of this technique.

Read vs. Write Operations

Before we plunge into the intricacies of the language card, let's distinguish between read and write operations. In BASIC, the statement X = PEEK(800) *reads* a value from (decimal) address 800 and assigns it to X. The statement POKE 800,128 *writes* the decimal value 128 into address 800. Examples of machine language read and write instructions are LDA $6000, which reads a value from (hex) address $6000 into the accumulator, and STA $6000, which writes the contents of the accumulator to address $6000. Reading is done implicitly when an instruction is fetched from memory; for example, JSR $6000 causes the instruction sequence beginning at address $6000 to be read and executed.

When more than one bank of memory with the same range of addresses is available, reading and writing to a bank are independent operations: it is possible to read from address $F000 in one bank but write to $F000 in another. In fact, when you power-up or reset your Apple, reading the range $D000 – FFFF will "look at" the motherboard ROM (Applesoft and the Monitor on an Apple] [+), while writing in the same range is done in bank 2 (described below) of the language card. If at this point you BLOAD a file to an address in this range, the file will be stored in the language card, but PEEK's to the loaded addresses will just return the contents of ROM. To make use of the file you just loaded, some bank-switching gymnastics are required.

How to Access the Language Card

The memory in the language card consists of addresses $E000 – FFFF, and two banks of memory $D000 – DFFF. Either bank can be enabled

```
:ASM
                    ********************************
                    *                              *
                    *       LC PAGE 2 BACKUP       *
                    *                              *
                    *        By Mark Harris        *
                    *           7/11/82            *
                    *                              *
                    * Call -A.P.P.L.E. : Dec. 1982 *
                    *                              *
                    * CALL BACKUP = $6000 to put   *
                    * picture (pg 2) in language   *
                    * card, OR RESTORE = $6021     *
                    * to restore picture to pg 2.  *
                    *                              *
                    ********************************

                            ORG  $6000

                    * Backup picture from page 2 to LC

6000: 2C 89 C0      BACKUP   BIT  $C089
```

NEW USES FOR THE RAM CARD

```
6003: 2C 89 C0   22           BIT  $C089      ;Write-enable LC (bank 1)
6006: 20 3B 60   23           JSR  INIT
6009: B1 FC      24   LP1      LDA  ($FC),Y
600B: 91 FE      25           STA  ($FE),Y
600D: C8         26           INY
600E: D0 F9      27           BNE  LP1
6010: A5 FD      28           LDA  $FD
6012: C9 5F      29           CMP  #$5F
6014: F0 07      30           BEQ  DONE
6016: E6 FD      31           INC  $FD
6018: E6 FF      32           INC  $FF
601A: 4C 09 60   33           JMP  LP1
601D: 2C 8A C0   34   DONE     BIT  $C08A      ;Read ROM, W protect LC
6020: 60         35           RTS
                 36
                 37   * Restore picture from LC to page 2:
                 38
6021: 2C 8B C0   39   RESTORE  BIT  $C08B      ;LC read (bank 1)
6024: 20 3B 60   40           JSR  INIT
6027: B1 FE      41   LP2      LDA  ($FE),Y
6029: 91 FC      42           STA  ($FC),Y
602B: C8         43           INY
602C: D0 F9      44           BNE  LP2
602E: A5 FF      45           LDA  $FF
6030: C9 FF      46           CMP  #$FF
6032: F0 E9      47           BEQ  DONE
6034: E6 FD      48           INC  $FD
6036: E6 FF      49           INC  $FF
6038: 4C 27 60   50           JMP  LP2
603B: A9 00      51   INIT     LDA  #0
603D: A8         52           TAY
603E: 85 FC      53           STA  $FC
6040: 85 FE      54           STA  $FE
6042: A9 40      55           LDA  #$40
6044: 85 FD      56           STA  $FD
6046: A9 E0      57           LDA  #$E0
6048: 85 FF      58           STA  $FF
604A: 60         59           RTS

--End assembly--  75 bytes    Errors: 0
```

```
]LIST

10  REM *** PICTURE
      MOVE ***
20  PRINT  CHR$ (4)"BLOAD
      LC PAGE 2 BACKUP"
30  HGR2 : HCOLOR= 3:
      HPLOT 50,50 TO
      250,50 TO 250,150
      TO 50,150 TO 50,50:
      REM DRAW RECTANGLE
40  CALL 24576: REM
      BACK UP PICTURE
50  HGR2 : REM  ERASE
      PICTURE
60  GET A$: REM WAIT
      FOR KEYPRESS
70  CALL 24609: REM
      RESTORE PICTURE
```

for reading and/or writing by referencing certain addresses. These are as follows:

	Bank 1 references. Read ROM	Read RAM
Write enable (2 references)	$C089 (–16247)	$C08B (–16245)
Write protect	$C08A (–16246)	$C088 (–16248)
	Bank 2 references. Read ROM	Read RAM
Write enable (2 references)	$C081 (–16255)	$C083 (–16253)
Write protect	$C082 (–16254)	$C080 (–16256)

For example, to read the Apple ROM (Applesoft and Monitor in the Apple] [+) and write-protect language card RAM, bank 1, the instruction STA $C08A could be used. (LDA instead of STA is OK, but loses the contents of the accumulator. BIT may also be used, but can change the status register flags.) In this state the instruction STA $D000 would have no effect, since no RAM with that address is enabled. To read ROM and write-enable bank 2 RAM, the pair of instructions BIT $C081, BIT $C081 would work. In this state, LDA $D000 would load the accumulator with the contents of that address in the motherboard ROM, while STA $D000 would store the contents of the accumulator in bank 2 RAM. Note: If additional code separates the two references to $C081, the same state will be set after the second BIT $C081 so long as no write-protecting takes place in between. In fact, whenever any two write-enables (even to different banks) take place without intervening write-protects, the effect will be the same as a double-reference to the last address.

The LC PAGE 2 BACKUP Program

LC PAGE 2 BACKUP is a short machine-language routine which will copy a high-resolution graph (page two) to or from addresses $E000 – FFFF in the language card. The entry point for backing up a picture is BACKUP = $6000 (24576 decimal), and that for restoring the picture is RESTORE = $6021 (24609). (Note that since the range $E000 – FFFF used in the program is unique in the language card, it doesn't make any difference whether bank 1 or 2 is enabled.) One use for the program would be to create an image on Hi-Res page 2, back it up, alter it and save the new graph to disk, then restore the original for further alterations. Of course the same objective could be accomplished using the Apple's own memory rather than that in the language card, but this only an illustration. The method demonstrated can be applied to many situations in which block moves are desirable.

How to Enter and Save Binary Files

The numbers on the far left of the assembly listing are hexadecimal addresses, and the numbers immediately to the right are the contents of those addresses. To enter, get into the Monitor (CALL -151) and type "6000:." Now type the contents, "AD 89 C0 AD 89 C0 20 ...." You can enter up to 85 bytes before overflow occurs; before you have typed this many, hit return and continue with a colon and more data. This entire routine could be entered by typing:

```
6000:2C 89 C0 2C 89 C0 20 3B
: 60 B1 FC 91 FE C8 D0 F9
: A5 FD C9 5F F0 07 E6 FD
: E6 FF 4C 09 60 2C 8A C0
: 60 2C 8B C0 20 3B 60 B1
: FE 91 FC C8 D0 F9 A5 FF
: C9 FF F0 E9 E6 FD E6 FF
: 4C 27 60 A9 00 A8 85 FC
: 85 FE A9 40 85 FD A9 E0
: 85 FF 60
```

To save the program, you need the starting address and length of the file. These are $6000 and $4B (75 decimal), respectively. Assuming you are back in BASIC (press [Ctrl-C] [return] from the Monitor) and have your disk drive ready, type "BSAVE LC PAGE 2 BACKUP,A$6000, L$4B." The following Applesoft program "PICTURE MOVE" illustrates the use of the routine you just saved by drawing a picture on Hi-Res page two, duplicating it in the language card, erasing page two, then restoring the picture. (See boxed listing on previous page).

Getting Fancy

The procedure just illustrated for moving data in and out of the language card is fairly straightforward. Locating subroutines in the added memory is more involved, but generally more useful. This is particularly desirable during program development, when utility routines can be placed in the extra memory without interfering with test runs of the program. In this application, the card need not be present for the finished product (the program being developed) to run. (I write most of my programs to run on Apples less fortunate than my own.) The new Program Global Line Editor, available from *A.P.P.L.E.*, uses the language card in this way.

If the subroutine to be placed in the language card contains no calls to ROM addresses $D000 - FFFF, the task of linking it to a driver program is easy. Assuming for example that the subroutine begins in bank 2 RAM at $D000, the calling (machine language) program could simply address $C083 twice to read-and write-enable the RAM, call the routine, then address $C082 to read the motherboard ROM and to write-protect RAM. To get your routine in the language card in the first place, you can BLOAD it there if the card has been write-enabled. The program OVERLAY illustrates this technique. When loaded at $D000 (bank 2 RAM) and called by the short driver OVERLAY DRIVER, this subroutine overlays the picture currently on high-resolution page 2 onto the picture on page 1 (without disturbing the contents of page 2). To try out the program, key in and BSAVE OVERLAY at a lower address, say $6000, then use the

```
:ASM
                    ********************************
                    *                              *
                    *          OVERLAY             *
                    *                              *
                    * A subroutine to overlay page *
                    * two picture onto page one.   *
                    * Load into language card and  *
                    * call from a machine language *
                    * program by                   *
                    *      STA $C080 ;Read bank 2  *
                    *      JSR $D000               *
                    * After return from subroutine *
                    * read ROM by STA $C082        *
                    *                              *
                    ********************************

                          ORG   $D000
                    *
                    PG1   EQU   $FC
                    PG2   EQU   $FE
D000:  A0 00              LDY   #0        ;Set up hi-res
D002:  84 FC              STY   PG1       ;pointers
D004:  84 FE              STY   PG2
D006:  A9 20              LDA   #$20
D008:  85 FD              STA   PG1+1
D00A:  A9 40              LDA   #$40
D00C:  85 FF              STA   PG2+1
D00E:  B1 FC        LOOP  LDA   (PG1),Y  ;"OR" corresponding
D010:  11 FE              ORA   (PG2),Y  ;bytes from page
D012:  91 FC              STA   (PG1),Y  ;one and two,
D014:  E6 FC              INC   PG1      ;store in page one
D016:  E6 FE              INC   PG2
D018:  D0 F4              BNE   LOOP
D01A:  E6 FD              INC   PG1+1
D01C:  E6 FF              INC   PG2+1
D01E:  A5 FD              LDA   PG1+1
D020:  C9 40              CMP   #$40
D022:  D0 EA              BNE   LOOP
D024:  60                 RTS
 --End assembly-- 37 bytes     Errors: 0

:ASM
                    ********************************
                    *                              *
                    *       OVERLAY DRIVER         *
                    *                              *
                    * Load OVERLAY at $D000 bank 2 *
                    * RAM, then BRUN this program. *
                    *                              *
                    ********************************

                          ORG   $6000
                    *
6000:  8D 80 C0           STA   $C080   ;Read bank 2 RAM
6003:  20 00 D0           JSR   $D000   ;Call OVERLAY
6006:  8D 82 C0           STA   $C082   ;Read ROM
6009:  8D 50 C0           STA   $C050   ;Display page
600C:  8D 52 C0           STA   $C052   ;two hi-res
600F:  8D 54 C0           STA   $C054   ;graphics
6012:  8D 57 C0           STA   $C057
6015:  4C D0 03           JMP   $3D0    ;Exit to BASIC

--End assembly--    24 bytes       Errors: 0
```

NEW USES FOR THE RAM CARD

A(ddress) parameter in BLOAD to move it to $D000 (see MESSAGE DRIVER below). Key in and BSAVE OVERLAY DRIVER at the addresses given in the assembly, then with pictures of your choice in Hi-Res pages 1 and 2, BRUN OVERLAY DRIVER. The combined images will be displayed on page 1. (Type "TEXT" to continue.)

Things get more complicated if the subroutine contains references to ROM $D000 - FFFF: an obvious conflict results. One way around the problem is to have the subroutine in the language card pass the address of the desired ROM routine to a short segment in primary RAM (below $D000), which in turn calls the routine and then returns to the language card. An assembly of such a segment, "INDIRECT LC CALL," is given below. If this code is placed at address $300, then a language card routine at $D000 (bank 2) can be called by a JSR $300, which will read-and write-enable RAM (bank 2) before entering the routine. The language card routine can in turn call (for example) the COUT subroutine (output a character) at $FDED in ROM by pushing the contents of the accumulator and the address $FDED onto the stack and executing a JSR $30D as follows:

```
PHA
LDA $FD
PHA
LDA $ED
PHA
JSR $30D
```

If you have a macro-assembler (such as Big Mac), you can define a short macro so that ">>> CALL COUT" will insert the above code during the assembly, thus making the bank-switching relatively invisible.

As an example of this indirect approach, let's try printing a simple message from a routine in the language card by calling COUT in the Monitor ROM. The BASIC program MESSAGE DRIVER will load such a routine, PRINT MESSAGE, into the language card and call it via INDIRECT LC CALL. (You must first BSAVE PRINT MESSAGE from a lower address, as was done with OVERLAY.) When MESSAGE DRIVER is run, the sentence "THIS IS A TEST." will print on the screen. Not very flashy, but any subroutine in Applesoft or the Monitor can be accessed in the same way.

```
]LIST

10  REM *** MESSAGE
      DRIVER ***
20 X =  PEEK ( - 16255):X
    =  PEEK ( - 16255):
    REM WRITE-ENABLE
    BANK 2 BUT READ
    ROM (POWER-UP
    STATE)
30  PRINT  CHR$ (4)"BLOAD
    PRINT MESSAGE,A$D000"
40  CALL 768: REM PRINT
    MESSAGE
```

```
:ASM

                        ********************************
                        *                              *
                        *      INDIRECT LC CALL        *
                        *                              *
                        *       By Mark Harris         *
                        *                              *
                        * Call -A.P.P.L.E. : Dec. 1982 *
                        *                              *
                        * JSR $300 to enter subroutine *
                        * at $D000 (bank 2) in lang/cd *
                        * ROM calls from LC routine    *
                        * must be done indirectly      *
                        * using the following code:    *
                        *                              *
                        *    PHA                       *
                        *    LDA #)SBR   ;High byte of *
                        *                ;SBR address  *
                        *    PHA                       *
                        *    LDA #(SBR   ;Low byte     *
                        *    PHA                       *
                        *    JSR $30D                  *
                        *                              *
                        ********************************

                                ORG   $300
                        *
0300: 8D 83 C0                  STA   $C083  ;Read RAM bank 2, write enable
0303: 8D 83 C0                  STA   $C083
0306: 20 00 D0                  JSR   $D000  ;Routine in l.c.
0309: 8D 81 C0                  STA   $C081  ;Read ROM
030C: 60                        RTS
030D: 68                INDCLL  PLA
030E: 8D 33 03                  STA   RTAD+1
0311: 68                        PLA
0312: 8D 34 03                  STA   RTAD+2
0315: EE 33 03                  INC   RTAD+1
0318: D0 03                     BNE   L1
031A: EE 34 03                  INC   RTAD+2
031D: 68                L1      PLA
031E: 8D 2A 03                  STA   SBR+1
0321: 68                        PLA
0322: 8D 2B 03                  STA   SBR+2
0325: 68                        PLA
0326: 8D 81 C0                  STA   $C081
0329: 20 FF FF          SBR     JSR   $FFFF  ;This adr gets changed to sbr originally called
032C: 8D 83 C0                  STA   $C083
032F: 8D 83 C0                  STA   $C083
0332: 4C FF FF          RTAD    JMP   $FFFF  ;Gets changed to return adr in lc program


--End assembly--        53 bytes    Errors: 0
```

I decided to really put this approach to the test by putting the RENUMBER program from the System Master diskette into the language card. It took a while, but it worked, and I now have a nice program-development tool which isn't clobbered by FP or by tampering with HIMEM, and which doesn't invade memory craved by the program I'm working on. Here is how I did it:

I first ran RENUMBER to get the machine-language program in its proper place in memory ($8E00 – 95FF). I then disassembled the routine using the program SOURCEROR from the Big Mac diskette (there are a number of such magnificent bonuses on the disk – I have found Big Mac to be an outrageous bargain). All external references were listed at the bottom of the disassembly, and the next task was to fix the ROM calls by the method described above. A couple of additional (but direct) steps were necessary because of a JMP rather than JSR to a ROM address in the routine. Finally, I changed the start (ORG) of the routine and re-assembled.

An Alternative Approach.

If the routine(s) placed in the language card make frequent calls to the Monitor in ROM, it may be worth putting a copy of the Monitor ($F800 – FFFF) into the card. This will speed up your program by eliminating bank-switching during Monitor calls. Only one-eighth of the card is sacrificed to this duplication of code, and the Monitor will be available in RAM as long as the power stays on and you don't write over it.

Conclusion

Chances are that you can get more mileage out of your language card than you've been getting. Practicing some of the techniques described above will suggest uses of your own. In Part 2 of this article, which will appear in our companion publication: *Call —A.P.P.L.E. in Depth*, issue 3: *All About DOS*, I'll present a disk-backup program that uses the language card to speed up the copy process.

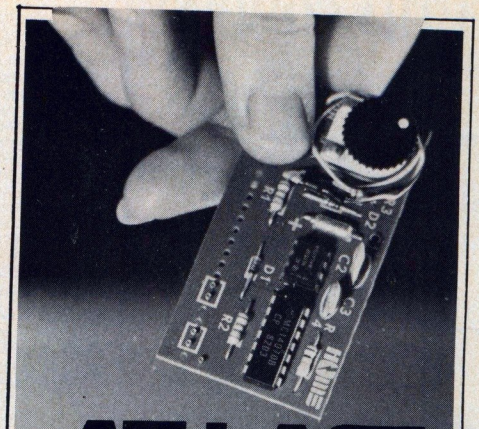
Bibliography

[1] *D. Worth & P. Lechner, Beneath Apple DOS, Quality Software, Reseda, CA 1981.*

[2] *W. Leubbert, What's Where in the Apple?, Micro Ink, Inc., Chelmsford, MA, 1981.*

(Both of the above are available to A.P.P.L.E. members at typical discounts.)


```
                        ********************************
                        *                              *
                        *          PRINT MESSAGE       *
                        *                              *
                        * Test program to call "COUT"  *
                        * from language card.  Call by *
                        * loading "INDIRECT LC CALL"   *
                        * at $300 and issuing a        *
                        * "JSR $300"                   *
                        *                              *
                        ********************************

                                ORG   $D000
                        *
                        COUT    EQU   $FDED
D000:  A0 00                    LDY   #0
D002:  B9 16 D0         NXTCHR  LDA   MSG,Y
D005:  F0 0E                    BEQ   RET
D007:  48                       PHA
D008:  A9 FD                    LDA   #>COUT
D00A:  48                       PHA
D00B:  A9 ED                    LDA   #COUT
D00D:  48                       PHA
D00E:  20 0D 03                 JSR   $30D
D011:  C8                       INY
D012:  4C 02 D0                 JMP   NXTCHR
D015:  60               RET     RTS
D016:  D4 C8 C9
D019:  D3 A0 C9
D01C:  D3 A0 C1
D01F:  A0 D4 C5
D022:  D3 D4 AE         MSG     ASC   "THIS IS A TEST."
D025:  00                       BRK

--End assembly--      38 bytes        Errors: 0
```

AT LAST
AUTO REPEAT & HIGH SPEED CURSOR
the REPEATERRRR™
WORD PROCESSORS, PROGRAMMERS, and VISICALC users have found it invaluable.
"The best Apple II keyboard enhancement for the money."
"Excellent documentation, very easy to install."
"LLLLove your Repeaterrrr!"

Apple II keyboard adapter
Repeats any keypress, including control characters for cursor moves, rubout, scrolling, etc. Standard feature on Apple III, IBM and most professional terminals. More convenient than Apple II's repeat key. On/off control with adjustable delay to match your typing touch.

high-speed repeat
TWO SPEED REPEAT, keyboard selectable. The best part. Use Apple's own repeat key to DOUBLE the repeat rate. Great for long cursor moves. Zip through text, program, or across a VisiCalc spreadsheet to get there FAST. A must for 80 column displays.

easy installation
Remove Apple top and plug in. Works with ALL other available Apple hardware. Fits late Rev. 6 (with encoder board) and all Rev. 7 or later keyboards.

no risk—30 day trial
Check, VISA, MasterCard (include card number and expiration date). Add $2 ($5 foreign) per order for shipping and handling. Ohio residents add 6½% sales tax. Full refund with 30 days. $19.95 introductory price until 11/30/82.

only $24.95 each
Dealer inquiries invited.

HOME
HIGHORDERMICROELECTRONICS CORP.
17 RIVER STREET
CHAGRIN FALLS OHIO 44022
PHONE 216-247-3110
APPLE IS A TRADEMARK OF APPLE COMPUTER, INC.
VISICALC IS A TRADEMARK OF PERSONAL SOFTWARE INC.

FUTRA COMPANY

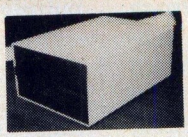
(213) 328-8951 (800) 421-5006
TWX 910 349 6211
AGENFTRA TRNC

Mail Order: Department C
P.O. Box 4380
Torrance, CA 90510-4380

Retail: OMC 20695 S. Western Ave.
Suite 124
Torrance, CA 90501
(213) 328-1760

apple computer
Sales and Service

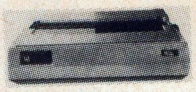
NEW FUTRA FLOPPY ADD-ON

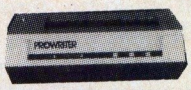


The new Futra 5¼" Floppy Disk Drive is fully software transparent with DOS 3.3 and 3.2.1 operating systems, as well as Pascal and CP/M. Add it to your present drive and controller for greater capacity and faster access. Just plug it in and go to work.
The Futra Apple-beige unit is built around a highly reliable Siemens system with over 10,000 lifetime hours. A shielded connecting cable is attached.
A Futra drive makes an excellent investment in preventing down time. At this price you don't have to be down a week waiting for your Apple drive to be fixed. Keep a spare Futra Drive just in case.
The Futra Drive features On-Track Head Seek with a precision lead screw to position the head onto the correct track. Retries and disk-to-disk copy errors are virtually eliminated.
Futra Drive has a faster Track Access Time at 25 MS than Disk II (40 MS), and a faster Average Access Time at 298 MS (Disk II at 463 MS).
The unit has a one-year warranty and MTBF (Mean Time Between Failures) is 8500 power-on hours. That's almost 3 years based on 8 hours per day.

PRINTERS

OKIDATA
- Microline 82
- Microline 84



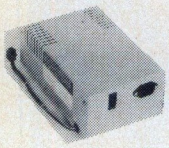


C. ITOH
- Prowriter
- Starwriter
- Printmaster

NEC 8023A

Grapplers Available

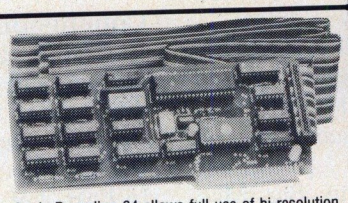
POWER



M & R Enterprises
SUP 'R' SWITCHER
$219.00
substitute supply

GE Surge Suppressor
Protect against power spikes ... **$7.50**

MICROTEK Apple Dumpling-64
Spooler/Buffer for Apple Computers

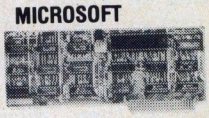


- Standard Dumpling GX **119.00**
- Dumpling-64 w/16K Buffer **179.00**
- Dumpling-64 w/32K Buffer **202.00**
- Dumpling-64 w/64K Buffer **225.00**

In addition to the standard graphics features, the Apple Dumpling-64 is the next logical extension to the graphic dump boards. Now you can get all the standard graphics features and up to 64K of buffer memory. Why tie up your computer waiting for your printer to catch up. Just dump the information into the buffer area and away you go. You're not just buying an interface you're buying time.

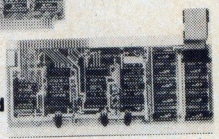
The Apple Dumpling-64 allows full use of hi-resolution functions:
Rotation, Inversion, Double Size, Emphasis Print, Dual Page Dump, Chart Recorder Mode and control over all variables for text and graphics.
Of course, there's PASCAL AND CP/M compatibility, ideal default conditions, and a standard 2 year warranty. Available for most graphics printers.

MICROSOFT



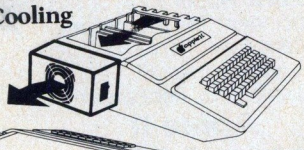
$ CALL $
Z-80 Soft Card with CP/M

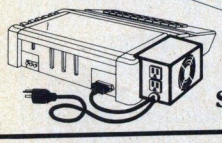
$ CALL $
16K Ram Card



Futra brings you the industry standard at it's lowest price ever. When you can get the trusted Microsoft quality, why buy anything else?

KENSINGTON MICROWARE
System Saver Fan $69.00
For Cooling





For Line Surge Suppression

COOL STACK – SENTRY II

$149.00

Protect your investment! In today's world of "Smash and Grab" it only takes a burglar a few seconds to make off with a computer that took you months or years to save for. Not only does the Cool Stack-Sentry II lock down, but it cools and organizes your system. Be safe not sorry.

ACRYLIC PLASTIC PRINTER STANDS

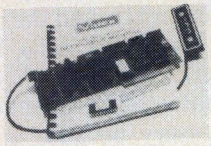
$22.50*

Put your printer on a tilted pedestal. Now you can check your printer with just a glance, and save space by storing your paper under the printer. Specify Epson MX-80, NEC 8023A, or Okidata Microline 82.

Epson MX-100 **$32.50***

*Available in Smoke Bronze

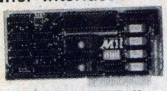
VIDEX INC.
VIDEOTERM
$ CALL $



When you're looking for a 80 column video board, why settle for anything less than Videx? Futra Co. makes it possible for you to own the best.

Manual Switch Plate	$19
Electronic Switch Plate	32.50
Micromodem II Firmware ROM	25.00

PRACTICAL PERIPHERALS
Buffer Interfaces



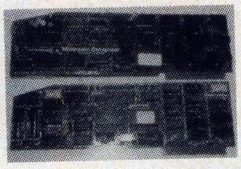
MICROBUFFER II INTERFACE

MICROBUFFER MBP-16K

Free your computer system from the binds of your printer!! Now you can get a buffer built into the interface that fits into the computer or a buffer that you install in the printer.

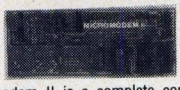
- Microbuffer II 16K **235.00**
- Microbuffer II 32K **279.00**
- MBS-8K Epson Serial (Printer) **149.00**
- MBS-16K Epson Parallel (Printer) **129.00**

MULTIFUNCTION CARDS



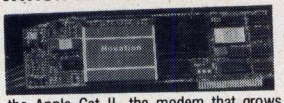
Mt. computer CPS Multi-function card **165.00**
Prometheus Versa Card **159.00**
Thunderware Thunderclock **105.00**

HAYES MICROMODEM II $289



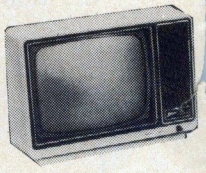
The Micromodem II is a complete communication system for the Apple II

NOVATION APPLE CAT II $269



Buy the Apple Cat II, the modem that grows with your needs. Find out about Novation's add-on features

Video Terminals & Monitors



Zenith 12"
$CALL$

BMC 12"

NEC:
- 12" Green Monitor
  12" Color Monitor (Composite)
  12" Color Monitor (RGB)

APPLE SOFTWARE – GAMES

Automated Simulations

Starfleet Orion	19.00
Invasion Orion	19.00
Temple of Apshai	29.00
Hellfire Warrior	29.00
Star Warrior	29.00
Crush, Crumble & Chomp	25.00
the Dragon's Eye	19.00
Jabbertalky	25.00
Sorcerer of Siva	25.00

Avante-Garde:

Hi-Res Computer Golf	26.00
Race for Midnight	26.00
Hi-Res Secrets	85.00
Zero Gravity Pinball	26.95
Death Race 82	26.95
Federation	26.95
Lazer Maze	26.95

Broderbund Software

Apple Panic	23.96
Space Quarks	26.95
Genetic Drift	26.95
Serpentine	31.50
Star Blazer	24.00
Track Attack	26.95
Red Alert	26.95
Midnight Magic	29.70
Arcade Machine	53.95
Choplifter	32.50
Labyrinth	24.50

Budgeco

Raster Blaster	23.96

Continental Software

LA Land Monopoly	26.95
Hyperspace Wars	26.95
3-D Skiing	22.45
Home Money Minder	29.70

Cal Pacific Computer

Ultima	31.96

Datamost

Swashbuckler	27.50
Casino	29.50
Snack Attack	24.50

INFOCOM

Zork I	32.50
Zork II	32.50
Deadline	44.95

Inovative Designs

Pool 1.5	27.96
Shuffel Board	26.95

Microsoft Consumer Products

Olympic Decathalon	23.96
Typing Tutor	19.96
Microsoft Adventure	26.95

Muse Software

Robot Wars	31.96
Three Mile Island	35.95
Best of Muse	35.95
Castle Wolfenstein	23.96
ABM	19.96

On-Line Systems

Jawbreaker	23.00
Threshold	34.50
Cannonball Blitz	27.96
Time Zone	89.96
Frogger	31.45
Mystery House	22.45
Marauder	31.45
Wizard & the Princess	22.45
Cranston Manor	31.45
Hi-Res Soccor	26.95
Ulysses & Golden Fleece	31.45
Soft Porn Adventure	23.96
H/R Football	35.95
Pegasus II	23.96
Missle Defense	23.96

Personal Software

Microchess 2.0	22.45
Checker King	22.45
Gammon Gambler	22.45
Bridge Partner	22.45
Monty Plays Monopoly	31.45
Monty Plays Scrabble	31.45

Piccadilly Software

Falcons	22.50
Warp Destroyer	24.50

Sirius Software, Inc.

Cops & robbers	31.45
Spoch	27.96
Orbitron	26.95
Gamma Gobblins	26.95
Sneakers	23.96
Escape from Rungistan	25.50
Kabul Spy	31.45
Bandits	27.96
Lemmings	24.50
Cyclod	24.50
Jellyfish	24.50
Minataur	24.50
Phantom Five	26.95
Space Eggs	26.95
Gorgon	31.96
Star Cruiser	22.45
Cyber Strike	35.95
Autobahn	26.95
Pulsar II	26.95

Sir-Tec

Wizardry	37.50
Knight of Diamonds	27.96

Strategic Simulation

Computer Bismark	42.00
Computer Ambush	42.00
Operation Apocalypse	42.00
Shattered Alliance	42.00
Computer Quarterback	27.00
Computer Conflict	27.00
Warp Factor	27.00
Cartels & Cutthroats	27.00
President Elect	27.00
Computer Baseball	27.00
Battle of Shiloh	27.00
Tigers in the Snow	27.00

Sub Logic

Flight Simulator	26.80
A2-PB1: Night Mission	23.96

FUTRA COMPANY GAME OF THE MONTH

Deadline	35.95

DATA SUPPLIES

Flip Sort
Plastic Storage case **$27.00**

Scotch Products: For reliable data recording use the best diskette for the money. With reinforced center!
Box of 10 - 5¼" storage carton ... **$27**
Scotch Head Cleaning Kit **$24**

MOUNTAIN COMPUTER

Apple Clock	219.00
ROM Plus	125.00
ROM Writer	141.00
CPS Multi-function Card	165.00
Music System	355.00
X/10 Control	160.00
Expansion Chassis	602.00

GENERAL APPLE SOFTWARE

Microsoft

Cobol 80	$562
Fortran 80	175
ALDS	95
TASC	149

Micropro

Wordstar	$235
Datastar	202
Mail Merge	85
Spellstar	129
Calcstar	130
Supersort I	159

Continental Software

Home Accountant	65.00
Home Money Minder	29.70

Software Publising Corp.

Personal Filing Systems	87.50
Personal Report Systems	85.50

PFS: Graph 109.00

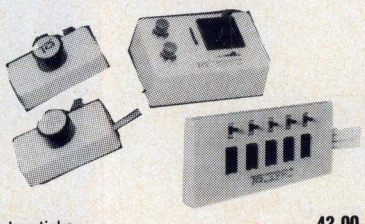
Visicorp

Visicalc	179
Visitrend/Visiplot	239
Visidex	211
Visiterm	125
Visiplot	169
Visischedule	253
Visifile	211
Visipack	549

Stoneware

DB Master	$179
DB Master Utility Pack	$89

TG PRODUCTS



Joysticks	42.00
Game Paddles	27.50
Selectaport	42.00

PADDLE ADDAPPLE $27.00

Game I/O over loaded? Don't open your Apple. Get the two-plug Paddle-Addapple. Switch selectable and reasonably priced.

TERMS: Shipping: add 3% for product shipped within continental USA via ups surface (minimum $3.00). **If the order placed is prepaid with U.S. funds in the form of check or money order, a Total Charge of $3.00 for shipping is all you pay within the continental USA vis UPS surface.** Allow 14 working days for personal and company checks to clear. Credit card charges limited to $1000. No COD's. FPO, APO or orders outside continental USA call or write for shipping charges or add 20% to purchase price (any difference will be refunded). CA residents add 6% sales tax (LA County 6.5%). All offers subject to change or withdrawal without notice.

*CPM is a registered trademark of Digital Research.

A PROGRAM TO CONVERT APPLEWRITER FILES

Thomas G. Williams

THE program given in the February, 1982, issue of *Call –A.P.P.L.E.* to convert Applewriter files to ASCII files requires both the original and converted files to be resident in memory at the same time. It cannot, as mentioned in the article, convert long (more than about 16000 bytes) files in one pass. Also, the loading of first Applewriter, then the binary program, and then an Applesoft program seems somewhat cumbersome. Applewriter files are loaded starting at $1900 (6400 decimal). If you just need your Applewriter files converted to ASCII without any formatting, there is plenty of space for a BASIC program to convert and store a new file. This article first describes a simple BASIC program to do this and then some machine language enhancements to speed up the process.

As noted in the previous article, Applewriter uses its own internal code, not standard ASCII, to store text. This code is designed for the display of text on the standard Apple display. Comparison of Applewriter binary files with the ASCII Screen Character Set Table on page 15 of the Apple] [Reference Manual indicates that an Applewriter file is formatted according to the following rules.

1. Files begin with a single control byte and end with a pair of control bytes. The initial byte is $83. The first of the pair of end bytes is $60, the second appears arbitrary.
2. A carriage return is stored in normal Apple ASCII ($8D).
3. Upper case characters are stored for display in inverse video and occupy the range [$00,$1F]. The ASCII character can be recovered by adding $C0 to the stored character.
4. Upper case numerals, punctuation marks, and special characters are accepted by Applewriter and stored in inverse video in the range [$20,$3F]. The ASCII character can be recovered by adding $80 to the stored character.
5. Lower case characters are stored as upper case normal video characters and occupy the range [$C0,$DF]. The ASCII character can be recovered by adding $20 to the stored character.
6. Ordinary numerals, punctuation marks, and special characters are stored in normal video in the lower case range [$E0,$FF]. The ASCII character can be recovered by subtracting $40 from the stored character.
7. The ranges from [$40,$7F], flashing characters; [$80,$9F], control characters; and [$A0, $BF], numerals, etc. are not used, except for carriage return and the beginning and end characters.

Therefore, we can use the following rules to convert character codes, taking advantage of rule 7 to reduce the number of code ranges that must be checked. If C is the stored code and P is the desired ASCII code, then

```
IF   0<=C< 32 THEN P=C+192
IF  32<=C<123 THEN P=C+128
IF 128<=C<224 THEN P=C+ 32
IF 224<=C<256 THEN P=C- 64
```

Listing 1 shows the BASIC program to convert an Applewriter file. The actual conversion is handled in the single statement at line 220 of the program. It uses the arithmetic value of logical relations, which is neatly explained in another article in the same issue of *Call – A.P.P.L.E.*

The rest of the program is just the usual housekeeping needed to get the file in memory and write the converted file. Lines 120 and 130 request the name of the file to be converted and prefix it with "TEXT.", in case you forgot. Line 150 checks the initial byte to verify that an Applewriter file was loaded. Line 160 computes the length of the file from data stored by DOS when the file was BLOADed. Lines 200–240 form a loop which replaces each character in memory with its ASCII equivalent. The rest of the program allows the converted data in memory to be written as either a binary or a text file. A subroutine starting at line 500 allows the insertion of carriage returns at user-specified intervals for use in other word processors.

]LIST Listing 1

```
10  REM

    APPLEWRITER  FILE CONVERTER
     ** BASIC ONLY VERSION **

BY TOM WILLIAMS AND VAL J GOLDING

  CALL -A.P.P.L.E. :: DEC. 1982

100 D$ =  CHR$ (4):BUFRADR = 256 *
     ( PEEK (110) + 5): TEXT : HOME

110  INPUT "APPLEWRITER FILE NAME
     ";FILE$
120  IF FILE$ = "" THEN  PRINT D$
     "CATALOG": GOTO 110
130  IF  LEFT$ (FILE$,5) <  > "TE
     XT." THEN FILE$ = "TEXT." +
     FILE$
140  PRINT D$"BLOAD"FILE$;",A";BU
     FRADR
145 FILE$ =  RIGHT$ (FILE$, LEN (
     FILE$) - 5)
150  IF  PEEK (BUFRADR) <  > 131 THEN
      PRINT "FILE NOT IN APPLEWRI
     TER FORMAT": GOTO 300
160 EA =  PEEK (43616) +  PEEK (4
     3617) * 256: REM

200  GOSUB 800: FOR I = BUFRADR TO
     BUFRADR + (EA - 1)
210 X =  PEEK (I): IF X = 141 THEN
     240
220 X = X - 64 + 96 * (X < 224) +
     96 * (X < 128) + 64 * (X < 3
     2)
230  POKE I,X
```

CONVERTING APPLEWRITER FILES

```
240  NEXT : REM

300  HOME : PRINT "   ...CONVERSIO
     N COMPLETED": VTAB 4: PRINT
     "  [B]  SAVE AS BINARY FILE"
     : PRINT "  [T]  SAVE AS TEXT
      FILE"
310  PRINT "  [L]  LIMIT LINE LEN
     GTH": PRINT "  [Q]  QUIT": PRINT
     : HTAB 8: INVERSE : PRINT "
     SELECT ";: NORMAL : PRINT "
     ";
320  GET SAVFILE$: PRINT SAVFILE$
     : PRINT : IF SAVFILE$ = "B" THEN
      PRINT D$"BSAVE"FILE$",S,A"B
     UFRADR",L"EA: PRINT : PRINT
     "DONE": END
325  IF SAVFILE$ = "Q" THEN  END

330  IF SAVFILE$ = "L" THEN 500
340  IF SAVFILE$ = "T" THEN  PRINT
     D$"OPENT."FILE$: PRINT D$"WR
     ITET."FILE$
350  FOR I = BUFRADR TO BUFRADR +
     (EA - 1)
360  PRINT  CHR$ ( PEEK (I));
370  NEXT
380  PRINT : PRINT D$"CLOSE": PRINT
     : PRINT "DONE": END : REM

500  INPUT "LINE LENGTH ?";LL$:LL
      = VAL (LL$): IF  NOT  VAL
     (LL$) THEN LL = 239
510  GOSUB 800: FOR I = BUFRADR TO
     BUFRADR + (EA - 1)
520  COUNT = COUNT + 1
530  IF COUNT > LL - 10 AND  PEEK
     (I) = 160 THEN  POKE I,141:C
     OUNT = 0: GOTO 520
540  NEXT : GOTO 300
800  FLASH : PRINT : PRINT " WORK
     ING... ": NORMAL : RETURN
```

The output from this program differs from that of the original program since it lacks the formatting that would be done by the PRINT routine of Applewriter. There are two essential differences:

1. The original program will interpret all text-embedded printer commands, such as !np for new page. This program simply converts them.
2. The original program is writing output to an imaginary printer with 132 columns. It will supply carriage returns as necessary so that no string of characters without a carriage return will be longer than 132 bytes.

The importance of the second difference depends on what you plan to do with the converted data.

Long strings will not affect storage by DOS, since sequential text file records can be up to 32767 bytes long, and Applewriter files are only 32000 bytes maximum. If you just need raw data conversion for use with another machine or word processor, this program will work fine. If you are creating files to be EXEC'ed, you need to watch the string length in either case, because either insertion of arbitrary carriage returns or lines in excess of 239 characters will cause errors. If you plan to use BASIC programs to operate on the converted data, you may utilize the line length limit subroutine.

The only problem I have with this program is the lack of speed. For instance, it took about 4.8 minutes to convert this article, which is about 8150 bytes long, and another 1.5 minutes to write the converted data to a text file. This indicates that converting a full Applewriter file would require about 19 minutes, with another 6 minutes required if a text file is written. The culprit is Applesoft's inability to do integer arithmetic. In particular, in lines 200–240, the index I is maintained in floating point format and converted to integer for the PEEK statement. The value PEEKed is converted from a one byte integer to floating point, and all comparisons and computations in line 220 are done in floating point. Finally, the resulting floating point value is converted back to one byte for the POKE statement in 230. The same requirement for conversion affects the speed of the output loop at lines 350–370. Since the data is handled one byte at a time, these operations are natural candidates for use of machine language subroutines.

Listing 2 shows machine language subroutines which can be used to perform the conversion, output and line length limitation. The routine starting at XLATE does the character conversion, and the routine labeled OUT does the output for a text file after the file has been opened and prepared for writing. Label LIMIT handles the assembly equivalent of BASIC subroutine 500. Although a machine language subroutine could be used to simply perform the character conversion done by line 220 in the BASIC program, it is just as simple to perform the entire operation of scanning and converting the file that is done by statements 200–240. XLATE uses the same rules described above for conversion, although the order of the tests is altered somewhat. The routine tests for the end of text byte ($60) to terminate processing, rather than using the file length, because it is simpler. Notice that no arithmetic is actually needed for character conversion.

The text file writer OUT simply picks up each converted character and sends it to DOS via the standard COUT routine. It also uses the $60

BYTES & PIECES

Happy Holidays

Apple Software • All Programs On Disk

	List	Ours
Bag of Tricks	39.95	29.56
Cytron Masters	39.95	29.56
General Manager 2.0	229.95	170.16
Graforth II	75.00	55.49
Graphics Magician	59.95	44.36
Home Accountant	74.95	55.46
Knight of Diamonds	34.95	25.86
Logo [Krell]	149.95	119.96
Maxell Disks [10]	55.00	31.00
Nibbles Away II	69.95	51.76
P.F.S.	125.00	92.49
P.F.S. Report	95.00	70.29
Sensible Speller	125.00	92.49
Snooper Troops 1 or 2	44.95	33.26
Star Maze	34.95	25.86
System Saver Fan	89.95	76.46
Tasc Compiler	175.00	129.49
Zork 1, 2 or 3	39.95	29.56
SPECIALS		
Replay Card	129.95	108.96
Visicalc	250.00	175.00
P.F.S. Graph	125.00	87.50
Arcade Machine	59.95	41.96
Screenwriter Pro.	199.95	139.96
Wizardry	49.95	34.96

C.O.D. • Money Orders • Certified Checks • Personal Check Allow 2 Weeks
N.Y.S. Res. Add 7.25% Sales Tax • Orders Under $100.00 Add P&H
$2.00 U.S., $2.50 Canada, U.S. Funds
Foreign, Charges Only Min. P&H $6.00

• SEND FOR FREE PRICE LIST #908 •
Source TCP637

BYTES & PIECES (516) 751-2535
Box 525 Dept. C • E. Setauket, N.Y. 11733

Listing 2

```
                 ********************************
                 *                              *
                 * Applewriter Char. Translator *
                 *                              *
                 *        by Tom Williams       *
                 *                              *
                 * Call -A.P.P.L.E. : Dec. 1982 *
                 *                              *
                 ********************************
                 *

                 *
                          ORG   $1000
                 *
                 *
                 BUFPTR    =     0
                 LNG       =     $2          ;Default line length
                 HBYTE     =     $11         ;Default HOB start of file buf
                 CR        =     $8D
                 SPACE     =     $A0
                 NDCHR     =     $E0         ;Default end of file chr
                 * ;$E0 is Applewriter I format after conversion
                 *

                 *
                 * Character translate routine
                 *

1000: A9 01      XLATE     LDA   #1          ;Set up initial address
1002: 85 00                STA   BUFPTR
1004: A9 11                LDA   #HBYTE      ;POKE TRAD +5, HBYTE
1006: 85 01                STA   BUFPTR+1
1008: A0 00                LDY   #0          ;Clear for indirect addressing
                 *
100A: B1 00      LOOP      LDA   (BUFPTR),Y  ;Get character
100C: 10 10                BPL   UC          ;Branch if upper case
100E: C9 8D                CMP   #CR         ;If [C/R], ignore
1010: F0 1C                BEQ   NEXT
1012: C9 DF                CMP   #$DF        ;Letter or otherwise?
1014: B0 04                BCS   LCNUM       ;Branch if otherwise
1016: 09 20                ORA   #$20        ;Lower case letter
1018: 90 12                BCC   STCHAR      ;Always
101A: 29 BF      LCNUM     AND   #$BF        ;Numeral,etc.
101C: B0 0E                BCS   STCHAR      ;Always
101E: C9 60      UC        CMP   #$60        ;Upper case, check for end
1020: F0 14                BEQ   DONE
1022: C9 1F                CMP   #$1F        ;Check for letter or otherwise
1024: B0 04                BCS   UCNUM       ;Branch if not letter
1026: 09 C0                ORA   #$C0        ;Convert letter
1028: 90 02                BCC   STCHAR      ;Always
102A: 09 80      UCNUM     ORA   #$80        ;Numeral, etc.
102C: 91 00      STCHAR    STA   (BUFPTR),Y
102E: E6 00      NEXT      INC   BUFPTR
1030: D0 D8                BNE   LOOP
1032: E6 01                INC   BUFPTR+1
1034: D0 D4                BNE   LOOP        ;Always
1036: 60         DONE      RTS
                 *
                 * Character output routine
                 *
1037: A9 01      OUT       LDA   #1          ;Set up initial address
1039: 85 00                STA   BUFPTR
103B: A9 11                LDA   #HBYTE      ;POKE OUT +5, HBYTE
103D: 85 01                STA   BUFPTR+1
103F: A0 00                LDY   #0
1041: B1 00      LOOP2     LDA   (BUFPTR),Y
1043: C9 60                CMP   #$60
1045: F0 EF                BEQ   DONE
1047: 20 ED FD             JSR   $FDED       ;Send character to DOS
104A: E6 00                INC   BUFPTR
104C: D0 F3                BNE   LOOP2
104E: E6 01                INC   BUFPTR+1
1050: D0 EF                BNE   LOOP2       ;Always

                 *
                 *
                 *
                 * Limit line length subroutine
                 *     by Val J Golding
                 *
```

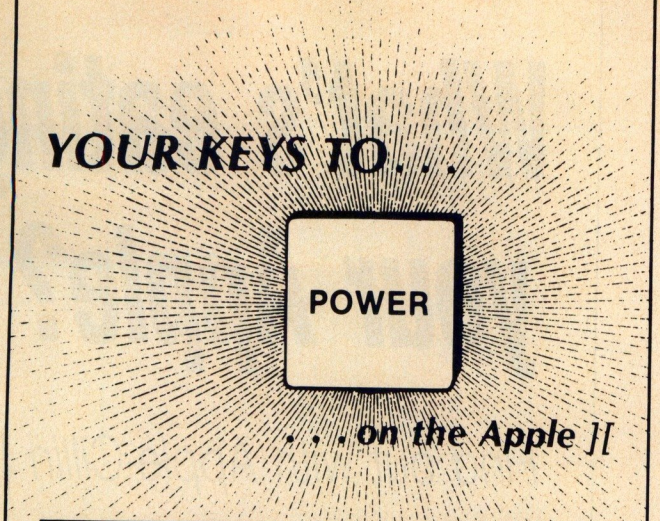
CONVERTING APPLEWRITER FILES

```
                  81
1052: A9 11       82   LIMIT    LDA  #HBYTE       ;POKE LIMIT +1, HBYTE
1054: 85 01       83            STA  BUFPTR+1
1056: A9 00       84            LDA  #$0          ;Lo byte
1058: 85 00       85            STA  BUFPTR       ;Setup for Y index reg
105A: AA          86            TAX               ;Initialize counter
105B: A8          87            TAY               ;Buffer index always zero
                  88   *
105C: E6 00       89   LOOP1    INC  BUFPTR       ;Bump buffer up
105E: D0 02       90            BNE  SKIP         ;if not zero, go to work
1060: E6 01       91            INC  BUFPTR+1     ;Bump hi byte
                  92   *
1062: E8          93   SKIP     INX               ;Inc char counter
1063: B1 00       94            LDA  (BUFPTR),Y   ;Get a char
1065: C9 E0       95            CMP  #NDCHR       ;End of file?
1067: F0 10       96            BEQ  EXIT         ;Yes, finish up
1069: E4 02       97            CPX  LNG          ;Reached end of line?
106B: 90 EF       98            BLT  LOOP1        ;Nope, let's try another
106D: C9 A0       99            CMP  #SPACE       ;Is it a space?
106F: D0 EB       100           BNE  LOOP1        ;Nope, let's get another chr
1071: A2 00       101           LDX  #0           ;Reset char counter
1073: A9 8D       102           LDA  #CR
1075: 91 00       103           STA  (BUFPTR),Y   ;Replace with a c/r
1077: 30 E3       104           BMI  LOOP1        ;Always taken
                  105  *
1079: 60          106  EXIT     RTS               ;That's it. Back to BASIC
    --End assembly--      122 bytes        Errors: 0
```

byte to terminate the process. LIMIT simply processes the file a second time, prior to calling OUT and writing to disk. It uses the X Register as a counter, then compares it against the user supplied line length, inserts a carriage return when called for and resets the character counter.

Listing 3 shows the BASIC program which incorporates the machine language routines. This routine simply loads the routines into free memory as indicated by the high byte of STREND, $6E(110), plus a safety margin. The file buffer is established similarly. The increase in speed of the conversion routine is dramatic: a full Applewriter file conversion requires about 1.5 seconds (instead of 20 minutes). In practice, this means that the conversion is done about the time that the disk stops spinning after the BLOAD. Sequential text file writing is speeded up by about a factor of 3.5.

```
]LIST        Listing 3

10  REM

   APPLEWRITER  FILE CONVERTER

BY TOM WILLIAMS AND VAL J GOLDING

  CALL -A.P.P.L.E. :: DEC. 1982

100 D$ =  CHR$ (4):BUFRADR = 256 *
     ( PEEK (110) + 5):BFILEADR =
     BUFRADR - 256:OUT = BFILEADR
      + 55::LIMIT = BFILEADR + 82
     :LNG = 2:NDCHR = LIMIT + 20:
      TEXT : HOME
105  PRINT D$"BLOAD TRANSLATOR,A"
     BFILEADR
107 HBYTE = BUFRADR / 256: POKE B
     FILEADR + 5,HBYTE: POKE OUT +
     5,HBYTE: POKE LIMIT + 1,HBYTE
110  INPUT "APPLEWRITER FILE NAME
      ";FILE$
120  IF FILE$ = "" THEN  PRINT D$
     "CATALOG": GOTO 110
130  IF  LEFT$ (FILE$,5) <  > "TE
     XT." THEN FILE$ = "TEXT." + FILE$
140  PRINT D$"BLOAD"FILE$;",A";BUFRADR
145 FILE$ =  RIGHT$ (FILE$, LEN (
     FILE$) - 5)
150  IF  PEEK (BUFRADR) <  > 131 THEN
      PRINT "FILE NOT IN APPLEWRI
     TER FORMAT": GOTO 300
160 EA =  PEEK (43616) +  PEEK (4
     3617) * 256: REM
200  CALL BFILEADR: REM XLATE
300  HOME : PRINT "  ...CONVERSIO
     N COMPLETED": VTAB 4: PRINT
     "  [B]  SAVE AS BINARY FILE"
     : PRINT "  [T]  SAVE AS TEXT
      FILE"
310  PRINT "  [L]  LIMIT LINE LEN
     GTH": PRINT "  [Q]  QUIT": PRINT
     : HTAB 8: INVERSE : PRINT "
     SELECT ";: NORMAL : PRINT "
     ";
320  GET SAVFILE$: PRINT SAVFILE$
     : PRINT : IF SAVFILE$ = "B" THEN
      PRINT D$"BSAVE"FILE$".S,A"B
     UFRADR",L"EA: PRINT : PRINT
     "DONE": END
330  IF SAVFILE$ = "L" THEN 500
340  IF SAVFILE$ = "T" THEN  PRINT
     D$"OPENT."FILE$: PRINT D$"WR
     ITET."FILE$
350  IF SAVFILE$ = "Q" THEN  END
360  CALL OUT
380  PRINT : PRINT D$"CLOSE": PRINT
     : PRINT "DONE": END : REM
500  INPUT "LINE LENGTH ?";LL$:LL
      =  VAL (LL$) - 10: IF  NOT
      VAL (LL$) THEN LL = 229
510  INPUT "ASCII VALUE FOR END O
     F FILE CHR          [DEFAULT
     = 224] ";EOF$:EOF =  VAL (EO
     F$): IF  NOT  VAL (EOF$) THEN
     EOF = 224
520  POKE NDCHR,EOF
540  POKE LNG,LL: CALL LIMIT: GOTO
```



zardax **Word processing** at its finest. Powerful and versatile, yet easy to use and natural. Designed for the business and professional environment (or for anyone who wants the best). Simple control commands. Typewriter-style shift and lock. Glossary. Form letters and mailing lists. Menus for disk access and printing. DOS 3.3 compatible text files. 40 or 80 column display. Modifiable drivers for most interface cards and printers. $295.

ZIP-COMM **Communications** add-on for ZARDAX. Turns ZARDAX into a communicating word processor, to send and receive text files. Talk to other Apples, mainframes, information services, typesetters. Includes terminal mode. 300 or 1,200 baud. Works with serial, modem, and popular 80 column cards used by ZARDAX. Log-on files and X-on X-off supported. $80.

AIDA "Apple Interactive Data Analysis." **Statistical analysis** package for production research work with large survey data files. Full range of analysis — from descriptive statistics to multiple regression. Complete data manipulation, transformation and case selection. Fast and accurate calculations. Up to 4,000 cases and 255 variables per file. $235.

Just push our button. . .

Action-Research Northwest
11442 Marine View Drive, SW.
Seattle, WA 98146
(206) 241-1645 Source: CL2542

Apple][is a trademark of Apple Computer, Inc.
ZARDAX is a trademark of Computer Solutions, Pty., Australia.

Dealer inquiries invited.

The Pizza Program

Announcing the first dinner menu planning system. It will save you time and add new zest to your meals. It may even convince your wife buying an Apple* was a stroke of genius.

ENDS HO-HUM DINNERS

Are you tired of the same old thing for dinner? Would you like more variety in your evening meal? Is there something you'd rather have but don't get very often? The Pizza Program is designed just for you. It's a delightful new software package designed to end the dinner-blahs with computer generated menus. Here is how it works.

You review what you like from the pre-selected food groups in the system. Delete any foods you don't enjoy. Add anything new at any time. Then decide how often you like to eat certain items. For example. Don't like liver? Then eliminate it with a few simple keystrokes. Or, you can plan for it as seldom as once every 99 weeks or as often as daily.

Want to go out to your favorite restaurant? Enter the restaurant's name as a "Main Course." Now your computer will automatically remind you to go out to eat — and as often as you select. It will delete all other items from that meal except the name of the restaurant.

AUTOMATIC SHOPPING LIST

You get a new menu each week or for just a few days if you want. And, it generates a detailed shopping list, automatically. It can arrange each item on the list in sequence according to the aisles at your favorite store. Studies show a shopping list will discourage impulse buying and save you money.

Also, it generates a per serving calorie counter. This is easy to delete anytime you are not in a diet mood or want to celebrate for any reason. You never count calories unless you want to.

RANDOMLY DELICIOUS

Say goodby to boring meals. Your computer will remember variety is the spice of life. This system makes eating at home a pleasure again. Each menu is randomly generated from major food groups according to the specific criteria you select. The system is easy to learn and easy to operate. Yet it is a sophisticated piece of software which will prevent menu mix-ups.

It will add a new dimension to your home meals without increasing your food costs. It is rumored The Pizza Program may actually reduce the climbing divorce rate. Husbands now have something to look forward to for dinner. Wives think it is terrific because it saves time and effort. And, kids love anything computerized. This is a useful and practical application you'll appreciate day after day, week after week.

TRY IT FOR 30 DAYS WITHOUT RISK

This tested system is guaranteed to make your life easier and happier. Our home trial lets you actually use The Pizza Program for a full 30 days before you decide to keep it. Watch the fun and convenience it creates. Enjoy better meals and see how much time it saves. If you are not satisfied for any reason, return it within 1 month for a prompt and courteous refund. Your investment is just $34.50 plus $2.00 for shipping and handling. (California residents add 6½% sales tax.) Full documentation is included.

OUT OF THE RUT

One housewife's reaction to this program is typical. She wrote, "Before using your system I found myself getting into a rut of serving the same things over and over. The Pizza Program has changed all of this for me. We now have a wider variety of dinners and best of all I don't have to decide what they will be. If this was all it did, I'd be thrilled. But it isn't. The shopping list I receive along with my menus has been such a time saver. I quickly run through it and delete anything I feel I don't need and add something I might. I would have a hard time going back to doing my menus by hand."

This system requires an Apple II Plus* with 48K and 1 disk drive. We urge you to take advantage of our no-risk, 30 day home trial offer. To order call toll free and use your VISA or MasterCard. Or, send a check to the address below. There's no obligation. Order today.

ORDER TOLL FREE
(24/hrs/day)
800-453-4000
(In Utah call **1-800-662-8666**)

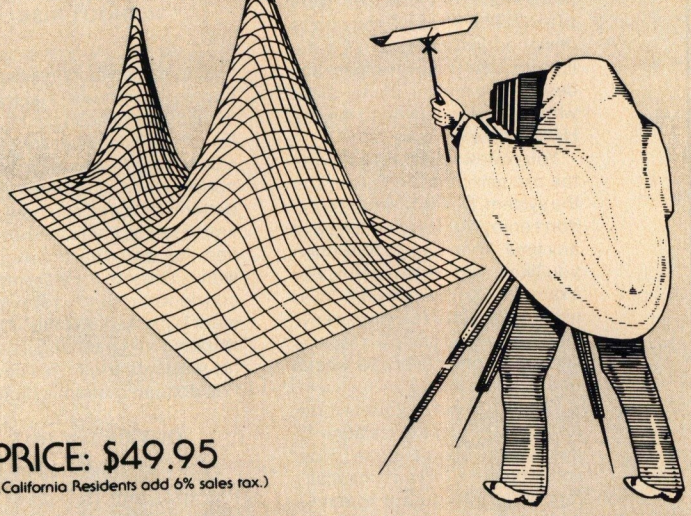
Gourmet Software
671 Eden Avenue
San Jose, CA 95117

*Apple is a registered trademark of Apple Computer, Inc.

TIRED OF ALL THE "EXCEPT FOR. . ."S?

WITH THE NEW PRINTOGRAPHER GRAPHICS PRINTING SYSTEM, WE'VE GOT YOU COVERED!

PRINTOGRAPHER™



PRICE: $49.95
(California Residents add 6% sales tax.)

The **PRINTOGRAPHER** is designed to fill all of your graphic printing needs, without having to worry about running into the problem of it almost working, "except on your printer", or "except for the lack of that particular feature". Whether you have a daisy wheel or dot matrix printer, the standard version of **PRINTOGRAPHER** works on any printer and interface combination with graphics capabilities. In many cases, this includes printers you may not even have thought could print graphics.

Just a FEW of the possible printers include: EPSON, PAPER TIGER, ANADEX, NEC, DIABLO, QUME, MPI, SILENTYPE, OKIDATA, MALIBU; interface cards include: APPLE, SSM, CCS, MTN COMP. CPS, MPI, GRAPPLER, TYMAC, PROMETHEUS and more!

In addition to versatile print options (easy cropping, variable magnifications, normal/reverse inking, vertical/horizontal format, etc.) **PRINTOGRAPHER** offers such unique features as the ability to print pictures directly from disk (without loading a file), spooling via our **DOUBLETIME PRINTER** package, or sending pictures over a phone line using **ASCII EXPRESS**. You can even put graphics in your text documents with our text editor software, **THE CORRESPONDENT**. As if that wasn't enough, we've made it easy to put the **PRINTOGRAPHER** routines right in your own programs to do Hi-Res printing immediately during their operation, without having to save screen images to disk!

We also know you see a lot of advertising these days for a truly overwhelming volume of software, all claiming to be the best, so we make this simple guarantee:

IF YOU CAN FIND A BETTER PACKAGE THAN (OR ARE AT ALL UNSATISFIED WITH) THE PRINTOGRAPHER WITHIN 30 DAYS OF PURCHASE, SIMPLY RETURN THE PACKAGE FOR A COMPLETE REFUND, NO QUESTIONS ASKED!

For more information, see your local dealer, or write **SOUTHWESTERN DATA SYSTEMS** for a free catalog. If your dealer is out of stock, we can ship **PRINTOGRAPHER** to him within 24 hours of a call to our offices.

REMEMBER: WITH PRINTOGRAPHER, YOU'RE PICTURE PERFECT!

southwestern data systems™

10761-E Woodside Avenue • Santee, California 92071
Telephone: 714/562-3670

What's eating your Apple?

Find out with Apple-Cillin II™

If you use your Apple for your business or profession, you probably rely on it to save you time and money. You can't afford to guess whether it is working properly or not. Now you don't have to guess. Now you can find out with Apple-Cillin II.

Apple-Cillin II is the comprehensive diagnostic system developed by XPS to check the performance of your Apple II computer system. Apple-Cillin II contains 21 menu driven utilities including tests for RAM memory, ROM memory, Language Cards, Memory Cards, DISK system, Drive Speed, Keyboard, Printer, CPU, Peripherals, Tape Ports, Monitors and more. These tests will thoroughly test the operation of your Apple, and either identify a specific problem area or give your system a clean bill of health. You can even log the test results to your printer for a permanent record.

Apple-Cillin II works with any 48K Apple system equipped with one or more disk drives.

To order Apple-Cillin II - and to receive information about our other products - Call XPS Toll-Free: 1-800-233-7512. In Pennsylania: 1-717-243-5373.

Apple-Cillin II: $49.95. PA residents add 6% State Sales Tax.

XPS

XPS, Inc.
323 York Road
Carlisle, Pennsylvania 17013
800-233-7512
717-243-5373

Apple II is a trademark of Apple Computer Inc.

VIDEO FOLLOW UP

Jim Hockenhull

RESPONSE to "Video Interfacing" in the June, '82 issue has been overwhelming. As a result of calls and letters three more related products have come to light. All claim to convert the Apple]['s weird video signal into full National Television System Committee (NTSC) standard, broadcast-quality output. They are:

The FULL NTSC CONVERTER,
BDL Elec
P.O.Box 43
Beloiel, Quebec
CANADA J3G 4S8
Price: ca. $800.

This unit consists of a printed circuit board which fits in Peripheral Slot #7 and a few dual-inline-package (DIP) headers, inserted between integrated circuits (ICs) and the Apple motherboard. A bit of soldering is also required for installation. The Converter requires an external composite video source for "gen-lock," a system of regenerating synchronizing pulses and color subcarrier so that the new Apple signal is "in step" with everything else. In a broadcast system it can be treated like a camera and used in fades, wipes, and keys.

Reports from users of the Converter have ranged from highly enthusiastic to very negative. A drawback mentioned by everyone is that colors tend to noticeably jitter and crawl. In the U.S., the Converter is distributed by

Broadcast Engineering
4973 Via Cinta
San Diego, CA 92122
(714) 543-6206

and

Photo & Sound Co.
19221 66th Ave South
Seattle, WA 98188

The VB-1 Computer Board,
Video Associates Labs, Inc.
2304 Hancock Drive, Suite 1-F
Austin, TX 78756
(512) 459-5684
Price: ca.$1850

The VB-1 consists of two boards, one of which fits into any of the Apple's peripheral slots, the other being mounted on top of the power supply. Other connections to the motherboard are made through DIP headers. No soldering or trace cutting is required. It also needs an external composite video source for synchronization; Apple graphics are "keyed" (superimposed) over this source. No computer memory is taken up by the board's operation; it is compatible with all Apple software.

On the market for some time, over 200 VB-1 units are now in use. Typical applications include the generation of color titles, of weather maps, logos, animations and other TV graphics.

The VB-3 MICRO-KEYER,
also from Video Associates Labs.
Price: ca. $2400

A newer and more versatile board than the VB-1, about 70 have been sold. The VB-3 will not only gen-lock to external video sources but can also stand alone with no video input. It incorporates its own full screen color matte generator, an analog device which will fill the entire screen with any of 2,560,000 colors (!) and key the Apple graphics over it. Saturation, hue, luminance, horizontal phase, and color-burst phase (and a few other things) are controllable through about 2.5K of driving software. Source listings of the software are provided.

In normal operation the VB-3 displays the usual undersized Apple screen (the rest, remember, can be filled in with a couple of million background colors). Interestingly, software is now under development to allow the Apple to generate a full screen (360x240) of graphics, combing Hi-Res pages 1 and 2.

Testimonials for both the Video Associates Labs units have been glowing, to say the least, one user describing the VB-3 as "as good as it can get." Mr. Saul Bernstein of Thousand Oaks, California recently won an Emmy for TV graphics done with an Apple/VB-3 combination.

I emphasize that I have not seen any of these products in operation; I'm merely passing along information gathered from manufacturers, distributors, and users. Apparently advertising has been done in the broadcasting journals and hence has escaped the notice of most us computerists.

And finally, a couple of follow-ups of the June article:

a) I have as yet been unable to re-combine the Amdek DVM red-green-blue- signal into composite video. As a color display device, though, the DVM/Color II combination is absolutely wonderful.

b) Manufacturers' addresses were omitted from the article. They are:

Adwar Video Corp.
100 Fifth Ave.
New York, NY 10011
(2122) 691-0976

Amdek Corporation
2420 E. Oakton St.
Arlington Heights, IL 60005
(312) 364-1180

Symtec, Inc.
15933 West 8 Mile
Detroit, MI 48235
(313) 272-2950

My deepest thanks to all those who have called or written about this whole video problem.

```
]LIST

5  HOME
10  FOR X = 1024 TO
      2039 STEP  INT
      ( RND (1) * 10)
      + 3: IF  RND (1)
      ) .5 THEN  POKE
      X,32: NEXT : GOTO
      10
20  POKE X,160: NEXT
      : GOTO 10
30  REM

SO WHAT DID YOU EXPECT?

DAVE BROWN : CALL -A.P.P.L.E
```

THOUSANDS OF CP/M SOFTWARE PACKAGES ARE GATHERING IN SAN FRANCISCO

Moscone Center
San Francisco
Friday-Sunday
January 21-23, 1983
Sponsored by

CP/M '83®

DIGITAL RESEARCH™
The Creators of CP/M

CP/M '83 is an International Exposition and Conference for the CP/M industry and CP/M users. The exposition portion of the event will be the largest presentation of CP/M based hardware and software ever assembled. Nearly three hundred companies using over six hundred exhibit displays, will showcase the full spectrum of application packages, development aids, peripherals, accessories, publications and services for CP/M based computers . . . making CP/M '83 the largest first year computer event ever.

CP/M '83's Conference and Seminar program will include noted leaders from the industry including Gary Kildall, President, Digital Research Inc.; Sol Libes, Editor, Micro Systems Magazine; Christopher Morgan, Editor-in-Chief, Byte and Popular Computing Magazines; Adam Osborne, President, Osborne Computer Corporation; Tony Gold, Founder CP/M Users Group and Lifeboat Associates; Ben Rosen, President, Rosen Research; Portia Isaacson, President, Future Computing; Maggie Cannon, Editor in Chief, InfoWorld; David Crockett, Senior Vice-President, Dataquest; and Gordon Eubanks, Vice-President Language Div., Digital Research. CP/M '83's conference, seminar and workshop program is designed for computer tradespeople and users. The individuals listed above, plus dozens of others, will conduct informative discussions exploring CP/M applications, technical information, development aids, venture capital programs and software distribution. End user workshops will show users how to get the most from their CP/M computer.

CP/M '83 is sponsored by Digital Research Inc., the Creators of CP/M. Over 650 different companies support CP/M and more than 5,000 companies produce CP/M application packages . . . All of these firms will be represented at CP/M '83 . . . so attendees will see and try out application packages for every profession, plus state-of-the-art programs for word processing, telecommunications, graphics and data base management. Also on display or as workshop subjects are development aids to help you program faster, plus hundreds of CP/M compatible products appropriate for your applications.

At CP/M '83 you can explore the entire world of CP/M under one roof and learn more in three days than you could in six months, any other way . . . and everything which is on display is for sale.

Registration is $10 for one-day exhibits-only ticket, or $20 for a three-day ticket which includes admission to workshops, seminars and conferences. Attendees can pre-register thru the mail or purchase tickets at the door. To pre-register, send check payable to CP/M '83 to Northeast Expositions, 822 Boylston Street, Chestnut Hill, MA 02167, Telephone 617-739-2000. Call or write for the schedule of seminars, workshops and conferences plus San Francisco hotel information. All persons attending CP/M '83 can obtain very low, special convention hotel rates.

For information on exhibiting or attending call or write Northeast Expositions.

Name _____

Company (if any) _____

Address _____

City _____ State _____ Zip _____

Telephone (Area Code) _____

Check applicable box—Make checks payable to CP/M '83. Payment must accompany ticket orders.

☐ Send me ____ (quantity) one-day exhibits-only tickets @ $10 each per day.

☐ Send me ____ (quantity) three-day exhibits and conference tickets @ $20 each.

☐ Send me seminar, conference & workshop schedule along with San Francisco hotel reservation information.

☐ My company may be interested in exhibiting, send me information.

APPLEMASH

Michael Thyng

USUALLY we get involved in something heavy, but this time let's just have some fun. Something has plagued me for years; take a minute and contemplate this, I'm sure you can handle it.

This should probably be listed in the "I wouldn't have expected that department." Enter this program:

```
10  A = 1
20  PRINT A
30  A = A + .01
90  GOTO 20
```

Type RUN. Do you notice anything unusual, anything you didn't expect? That is the only worm in my Apple that I couldn't live with. A call to the Hot Line gave me the idea for a solution. Here is the algorithm for two decimal places.

```
40  A$ = STR$ (A)
50  FOR J = 1 TO LEN (A$)
60  IF MID$ (A$,J,1) = "."
    THEN 70
65  NEXT J
70  IF LEN (A$)    J + 2 THEN
    A$ = LEFT$ (A$,J + 2)
80  A = VAL (A$)
90  GOTO 20
```

That's probably worth the price of your membership right there.

```
]LIST

1   REM

SO WHAT DID YOU EXPECT

DAVID LOWE : CALL -A.P.P.L.E


10   HGR : HOME : DIM
       C(6)
20   DATA  96,17,10,4,10,17
30   FOR I = 0 TO 5:
       READ C(I): NEXT
       I
40   FOR J = 0 TO 26:
       FOR I = 0 TO 5
50 Y = J * 6 + I: POKE
       228,C(I)
60   HPLOT 0,Y TO 279,Y
70   NEXT I: NEXT J
80   VTAB 22: PRINT "SO
       WHAT DID YOU EXPECT?"
90   END
```


Coming Soon!

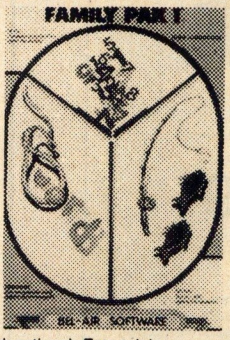
A.P.P.L.E
Computer Classes
beginning
January 17, 1983

CALL **Patti Hoffmann**
(206) 271-4514
A.P.P.L.E. Main Office
for more information

HAVE FUN WITH YOUR APPLE

Get to O'Hare International with all your clothes on. **$39.95**

Stop the Russian star fleet from destroying earth. **$29.95**



Educational, Entertaining, 3 games in 1. **$29.95**

VISA/MASTERCARD ACCEPTED

BEL-AIR SOFTWARE

P.O. Box 14414 Chicago, Illinois 60614
(312) 929-7727

Home Entertainment Division of Cortland Data Systems
Requires Apple II 48K. Apple is a trademark of Apple Computers, Inc.

Season's Greetings From A.P.P.L.E.

Apple PugetSound Program Library Exchange

"The world's largest Apple computer user's group."

NEW FOR YOUR APPLE™:

THE ULTIMATE GAMES FOR BIG BOYS.

AT AN INTRODUCTORY PRICE OF JUST $199.

These are the ones you play for KEEPS. They include everything your Apple needs to track commodities, trade stocks, compare options and either win or lose a fortune. Why go hunting for a measly pot of gold when you can play for stakes like that?

Until now, you had to ante up about $1,800.00 just to sit in on the game — only to find that price also bought you a lot of stuff you'd seldom use. But here's everything you need — not just to get started, but to do very sophisticated analysis and price management (see box). For only $199.

Now you can afford to track all that Haloid stock Aunt Millie promised to leave you. And even if you decide to buy every program in our library (over 30 now, and growing) you'll spend far less than $1800 for the most esoteric analysis possible.

Of course, if you're new to the investing game, you may choose to just dip in a toe, and pretend for a while. See if you really have the makings of a mogul. Ater all, if you take the plunge, you have nothing to lose except your car, your home, your family, and your Apple. Now that could be a REAL Adventure.

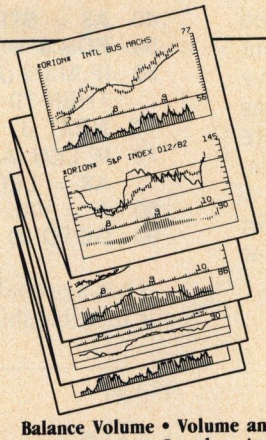
MasterCard and Visa holders order toll-free: **1-800-835-2246**

Ask for the CHART TRADER + PLUS package *(Demo not available through 800 number.)*

THE INVESTOR'S TOOLKIT

Our introductory CHART TRADER + PLUS package is $330-worth of software for $199.

It includes AUTORUN that lets you set up a procedure, then walk away to dinner while it loads data, does your studies, creates and prints charts all before you've had dessert. CONTAINS: • Single or Multiple Moving Averages • Overbought/Oversold Indicator • High/Low Price Band • Moving Average Oscillator • % Price Band • Constant Price Band • Daily Open, High, Low, Close Bar Chart • Graphs of 10 to 240 Days • Grid Lines for Update • Relative Strength Index • On Balance Volume • Volume and Open interest Graph • And more!



Includes Complete Data File Management — compatible with most stock and commodities data banks for telephone acquisition. Send $5 for demo disk, or call 312-648-1904 for more information.

OMEGA MICROWARE, INC.™

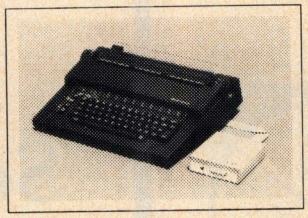
222 SO. RIVERSIDE PLAZA
CHICAGO, IL 60606
312-648-1904

Software by
Orion Management, Inc.

Data files are CompuTrac compatible

Apple is a registered trademark of Apple Computer Inc. Investor's Toolkit and Omega MicroWare, Inc. are trademarks of Omega MicroWare, Inc. © 1982 Omega MicroWare, Inc.

daisy quality printing ... $249



Now you can turn your Olivetti Praxis-30 or -35 daisy wheel typewriter into a letter quality printer. The **OLIVE-1** provides your Praxis with a microprocessor-based, serial communications interface (RS-232C) compatible with most computers. It features a wide range of options and supports all typewriter functions, including the entire printable character set (KBI & KBII). Complete with interconnect/installation instructions and user guide. Wall socket plug-in power supply (9VDC/500mA) optional. Also available: **OLIVE-80**, a fully integrated Praxis-30 plus **OLIVE-1** typewriter/printer package for under $750.00.

FEATURES

- ***Hardware Handshake***
- ***Software Protocols*** (ETX/ACK or XON/XOFF)
- ***Selectable Baud Rates*** (110, 300, 1200)
- ***165 Character Buffer*** (full line at 15 cpi)
- ***Printing Pitch Selection*** (10, 12 & 15 cpi)
- ***Keyboard I & II Selection***
- ***Control over Praxis buzzer***
- ***RS-232C Configuration Options*** (TXD/RXD crossing; Carrier Detect Assert; Handshake via CTS or DTR)
- ***Paper-Out Indicator LED***
- ***Reset & Resume-Print Switches***
- ***Printing Speed: Approx. 12 cps***

The olive branch Association, Ltd.

26291 Production Ave., Ste. 205
Hayward, CA 94545, USA
(415) 887-5633

OLIVE-1 & OLIVE-80 are trademarks of The Olive Branch Association, Ltd

SAVE $15 **EXPIRES JAN. 31, 1983**

For A.P.P.L.E. Members Only

WORDWEAVER /// SPECIAL

by Bob Huelsdonk

An Easy To Use Sophisticated and Comprehensive Word Processor For The Apple ///*

- Edit Commands Based on Program Line Editor
- Global Find and Replace
- Up to 253 Character Lines
- Imbedded Printer Controls Allowed

Regular Price: **$75.00**

SPECIAL $60.00

A.P.P.L.E. Orders
304 Ave. S., Suite 300
Renton, WA 98055
(206) 271-4514

Overseas shipping charge: $18.00
No other orders in the same envelope.

Washington state residents add 7.0% sales tax.

*Apple /// is a registered trademark of Apple Computer, Inc.

HOLIDAY SPECIAL
$100 OFF
SYSTEM PRICE
THROUGH
DEC.
31

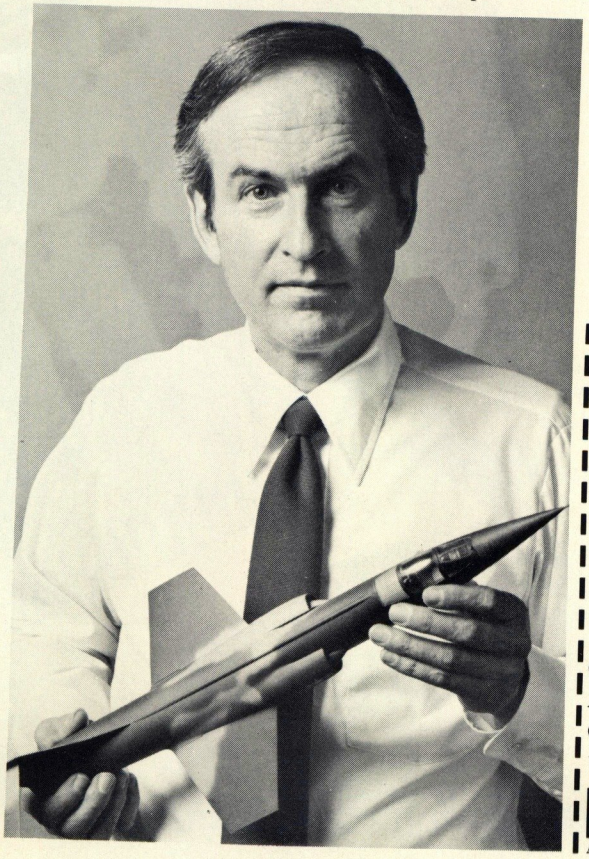
TEST-FLY A $20 MILLION JET ON AN APPLE? YES. WITH MICROSPEED.

At the Bethesda Naval Research Center, they've discovered the power of MicroSPEED. The Navy's engineers use this remarkable hardware/software combination to "fly" an advanced fighter aircraft in *real time*—even making vertical landings on a simulated carrier deck. A "crash" is merely another learning experience, and an opportunity to modify the research aircraft—inside the Apple—to improve tomorrow's combat planes.
Surprised that such a sophisticated task is possible on the Apple? So were the Navy's officials, and many others who have discovered...

THE MICROSPEED DIFFERENCE This extraordinary Language System exploits the real potential of the microcomputer for the first time. The difference between MicroSPEED and other programming languages is that with MicroSPEED, there is virtually *no limit* to what you can achieve. It may well be the ultimate language for the Apple II and III (and soon the IBM Personal Computer). MicroSPEED literally combines the performance of a *minicomputer* with an exhaustive set of user-friendly capabilities: hardware math processing, fast hi-res graphics and text, turtle graphics, print formating, two text editors, unlimited data types, and incredible FORTH extensibility—all at speeds up to 100 times faster than Basic.

USER-FRIENDLY, EASY-TO-LEARN Starting with simple commands that are comfortable even for non-programmers, MicroSPEED extends and builds, allowing you to create your own tailored application languages. The capability of your computer will grow exponentially, as you work in an active partnership with the machine, exploring and developing new problem-solving facilities—creating, correcting, refining your increasingly powerful system.

DEMANDING JOBS AT LOW COST Developed by a team of standout computer professionals, MicroSPEED has been put to the test in fields as diverse as medicine, the stock market, oceanography, and the arts. In even the most challenging applications, MicroSPEED users have been unanimous in their praise of the System and manual. Typical comments are:



"Very high marks,"
Thomas Tosch Phd., Tosch Information Management.

"The more I use MicroSPEED, the more I love it,"
James L. Hockenhull, University of Washington.

"Great!...A joy to use,"
Henry Harris, Mission Designer, Cal Tech's Jet Propulsion Lab.

"If you plan to use the Apple or IBM Personal Computer for <u>any</u> demanding task, then we built MicroSPEED for <u>you</u>,"
Sam Cottrell, President of Applied Analytics.

MicroSPEED requires the Apple Computer with single disk. MicroSPEED II includes 2 MHz math processor. MicroSPEED II+ includes 4 MHz math processor.

Applied Analytics Incorporated
8910 Brookridge Drive
Upper Marlboro, Maryland 20772 (301) 627-6650

Please send me:

_____MicroSPEED II, ~~$495.00~~ $395.- _____160 Page Manual, $15.00

_____MicroSPEED II+, ~~$645.00~~ $545.- _____Detailed Information

Name:_____

Company_____

Address_____

City_____State_____Zip_____Phone No.( )_____

Use this coupon to order, or for more information.

MicroSPEED
APPLE IS A TRADEMARK OF APPLE COMPUTER INC.

PICK A CARD, ANY CARD

AND GET THE BEST VALUE

AP-80 *APPLE II
Parallel Printer Card

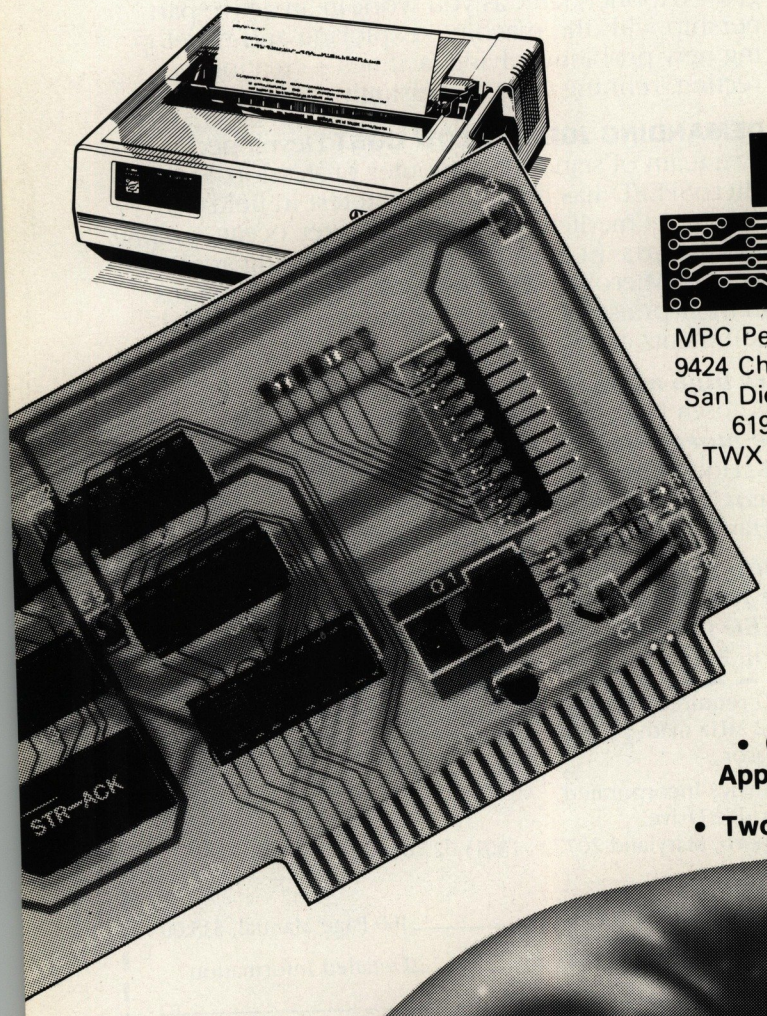
- Defaults to 80 column/printer only
- Conforms to Apple Control I Convention
- Compatible with Epson, Okidata and most other Centronics compatible printers.
- Compatible with all graphic dump routines
- Complete with cable, connector, and configuration module

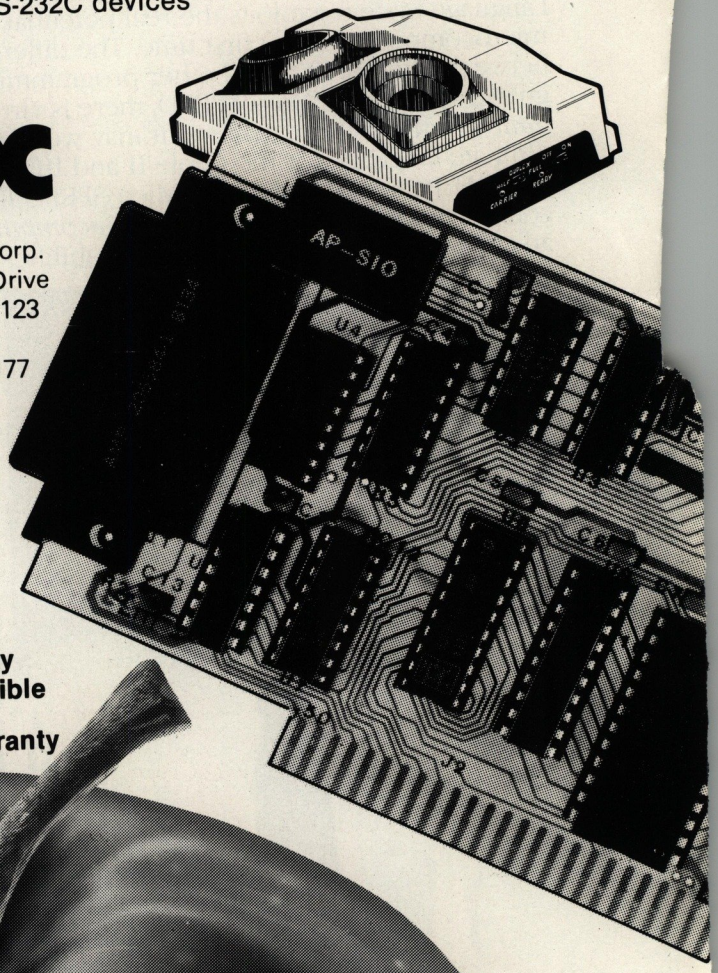
$89.50

AP-SIO APPLE II
Serial Input/Output Card

- Switch selectable crystal controlled baud rates from 50 to 19.2K baud
- Combines features of the APPLE High Speed serial and communications card
- Serial configuration modules for all serial communication protocols
- Switch selectable options include:
  1) Line feed on/off after carriage return
  2) Half/full duplex operation
  3) Ignore/not ignore incoming line feeds
- Compatible with most RS-232C devices

$129.50

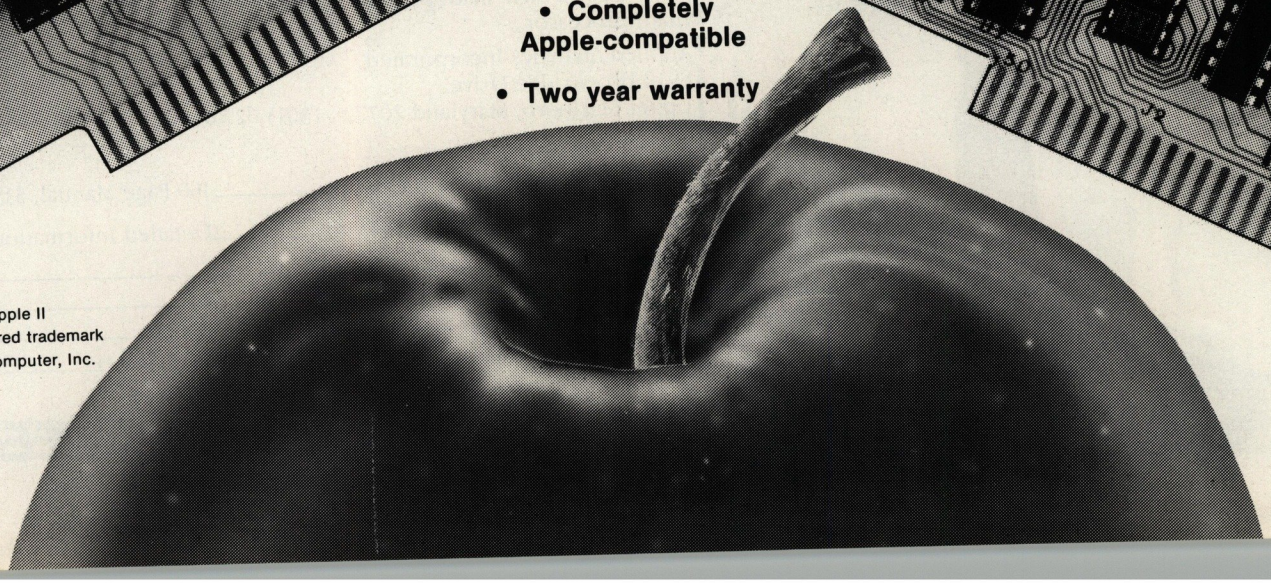




mpc

MPC Peripherals Corp.
9424 Chesapeake Drive
San Diego, CA 92123
619-278-0630
TWX 910-335-1177

- **Completely Apple-compatible**
- **Two year warranty**



*Apple II is a registered trademark of Apple Computer, Inc.

NEXT!

Wes Huntress

UNLIKE Integer Basic, in Applesoft it is not necessary to include an argument for the NEXT statement at the end of a FOR...NEXT loop. NEXT works just as well as NEXT I, and is even a bit faster. What the manual doesn't say, however, is that you can get in trouble when not using the argument if you are in the habit of prematurely terminating FOR... NEXT loops. Conditional termination of FOR...NEXT loops is a useful programming technique, and in Applesoft it is necessary to include the argument for the terminal NEXT statement in such loops. The accompanying demo illustrates what happens in conditionally terminated FOR... NEXT loops when the argument is omitted.

In formatting the demo program for *Call -A.P.P.L.E.*, several features of the REM statement were noticed. Applesoft formats REM statements according to four simple rules:

1. There is always two spaces between the line number and REM.
2. There is always one more space than you originally typed between REM and the first letter of your remarks.
3. The right margin is at column 32.
4. After the first line, the left margin is at column 5.

Given these rules, it is easy to format your Applesoft REM statements. First POKE 33,33 to set the right margin at column 32. Type the line number, followed by three spaces, REM, and one space less than you wish to have between REM and the first letter of your remarks. Now type your remarks to the end of the first line (column 32). If a word overruns the right margin, back up and space over it or hyphenate it appropriately. Use spaces as necessary at the right margin until the cursor moves to the next line at the far left margin (column 0). Use ESC A to edit over to column 5 and begin typing again. Repeat these procedures for each new line in order to right and left justify your remarks between columns 5 and 32.

```
]LIST

10   REM

     APPLESOFT  "NEXT" DEMO
        BY  WES HUNTRESS

CALL -A.P.P.L.E. : DEC. 1982
25   HOME : PRINT : PRINT "ST
     ART": PRINT
30   FOR I = 1 TO 5
35   :: FOR J = 1 TO 10
40   PRINT : PRINT "LINE 50"
45   :: NEXT
50   PRINT : PRINT "LINE 50"
55   NEXT
90   PRINT : PRINT "EXIT"
95   END : REM
99   REM

HOW MANY PRINTED LINES
DO YOU EXPECT?
TRY CHANGING LINE 45 TO
:: NEXT J, AND LINE 55
TO NEXT I. NOTICE
A DIFFERENCE?
```

```
]LIST

10   REM

SO WHAT DID YOU EXPECT?

TIM JACKSON : CALL -A.P.P.L.
      E.
15   TEXT : HOME : FLASH
       : PRINT "WARNING"
20   NORMAL : PRINT "REMOVE
       DISKETTES FROM
       YOUR DRIVES BEFOREYOU
       CONTINUE. THEN
       PRESS ANY KEY."
30   POKE  - 16368,0:
       WAIT  - 16384,128:
       POKE  - 16368,0
40   HGR2 : ROT= 0
50   FOR K =  PEEK (78)
       TO 0 STEP  - 3
60   POKE 232,K
70   FOR J = 208 TO 255
80   POKE 233,J:N =
       PEEK (K + J *
       256)
90   FOR S = 0 TO N:
       SCALE= S + (S
       = 0)
100  FOR H = 1 TO 6
110  HCOLOR= H: DRAW
       S AT 139 + H,96
120  NEXT H,S,J,K
```


TWO UNIQUE HI-RES SYSTEMS!

THE ANIMATOR A system for producing **animated 'film' strips** with a minimum of work. You enter **only key frames** - The animator calculates the in-between frames to produce smooth motion. There is capacity for 50 independently moving parts in each frame. **The Animator** contains everything you need to create **animated games, choreograph dances,** make **logos,** or just **create cartoons.**

★Demo includes ballet sequence (ballerina has 12 independently moving body parts).
★Over **50 pages** of documentation, including **3 tutorials** & directions for adding strips to your own programs.
★Powerful film editor includes: insert, delete, merge, translate, freeze frame, and many more.
★★**BONUS!** A system for making shapes/shape tables - works independently of **The Animator.** It contains some unique features such as automatic construction of shapes at user defined angles.

SIGHT 'n SOUND Enhance any stereo or cassette recording with a **fully synchronized Hi-Res** light show. Use designs of your own creation or use those supplied. Features:

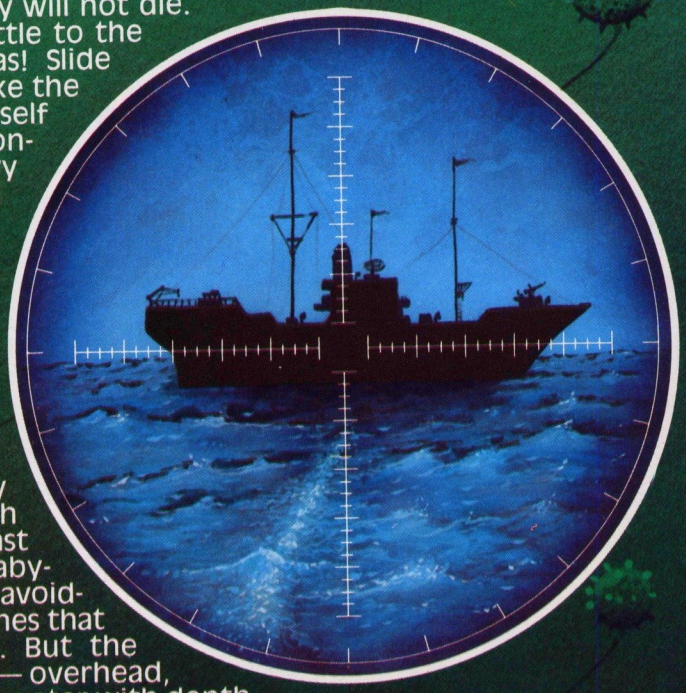
★Kaleidscope light show ★Guitar player who sings & plays to the beat ★Geometric design light show.
★21 page clearly written manual tells:
•How to create light shows
•How **SIGHT 'n SOUND** works
•How to add light shows to your own programs
★Watch your voice or 'EKG' on the oscilloscope
★9 predesigned light shows - ready to run
★Powerful design editor for creating light shows
★No hardware modifications.

Both systems require a 48K Apple II+ or II with Applesoft and a disk drive (DOS 3.2 or 3.3)
Send check or money order to
BalbeSoftware Systems
#6 White Plains Dr.
St. Louis, MO 63017

THE ANIMATOR $49.95
SIGHT 'n SOUND $24.95
Please add $2 ($7 overseas) postage/handling
For information call (314) 532-5377
Dealer inquiries invited
Apple is a trademark of Apple Computer, Inc.

ABSOLUTELY RELENTLESS

The challenge of inner space — the fury of an enemy that seemingly will not die. This is SEA DRAGON — a battle to the death **under** the high seas! Slide into the Captain's chair, take the controls and prepare yourself for the most incredible non-stop action this side of Davy Jones' locker. SEA DRAGON puts you in control of a nuclear sub that's armed from stem to stern with enough firepower to take on King Neptune himself — and you'll need every missile, every torpedo, and every scrap of skill you can muster to survive.

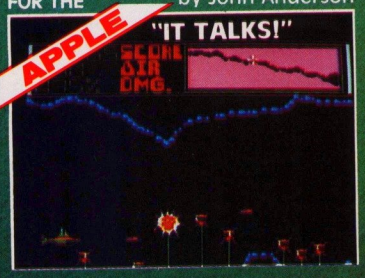


The object of SEA DRAGON is to successfully navigate your sub through an underwater course past mountains and through labyrinthine passageways while avoiding clusters of explosive mines that rise from the seabottom. But the danger doesn't stop there — overhead, surface destroyers lace the water with depth charges; below, deadly attack bases and arcing lasers cut a killing swath that could reduce your sub to bubbling slag. But even these potentially lethal perils are dwarfed by the awesome menace that awaits you at the course's end.

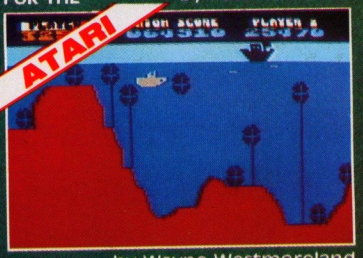
SEA DRAGON — every possible "extra" is here to ensure your playing pleasure: exciting sounds, high score save, machine language graphics and an eye-popping scrolling seascape that extends the equivalent of over two dozen screens placed end-to-end, providing a diverse and unique challenge that will not diminish after repeated playings.

Nothing you've ever seen on your micro could possibly prepare you for this! You are ready now, ready for the ultimate in undersea action with a pace that is absoutely unyielding. SEA DRAGON — the arcade has finally come home.

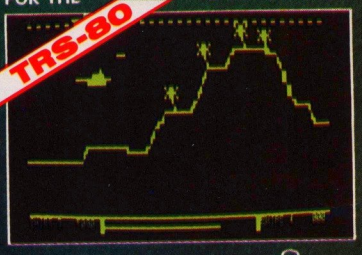
FOR THE APPLE by John Anderson



FOR THE ATARI by Russ Wetmore



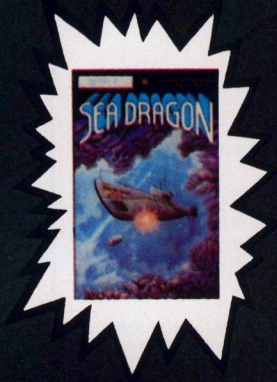
FOR THE TRS-80 by Wayne Westmoreland & Terry Gilman



ALL VERSIONS ARE JOYSTICK COMPATIBLE!

SEA DRAGON FEATURES

- Fantastic Scrolling Seascape
- Nearly Limitless Game Challenge
- High Score Save (disk version)
- Terrific Sound Effects
- Arcade Action Graphics™
- Apple version "talks" without special hardware!



ORDERING INFORMATION

APPLE 2 or APPLE 2 PLUS 48K Disk (DOS 3.3 required)
042-0146 ..... $34.95

ATARI 32K Disk
052-0146 ..... $34.95

ATARI 16K Tape
051-0146 ..... $34.95

TRS-80 32K Disk
012-0146 ..... $24.95

TRS-80 16K Tape
010-0146 ..... $24.95

Adventure INTERNATIONAL
A DIVISION OF SCOTT ADAMS, INC

PRICES SUBJECT TO CHANGE

To order, see your local dealer. If he does not have the program, then call 1-800-327-7172 (orders only please) or write for our free catalog.

Published by ADVENTURE INTERNATIONAL
a subsidiary of Scott Adams, Inc.
BOX 3435 • LONGWOOD, FL 32750 • (305) 830-8194

G HATTAWAY

EXPLORING THE WORLD OF APPLE:

DISK MAPPING THE Z-80 CP/M SYSTEM

Art Messler

THE Z-80 Softcard (by Microsoft) must be one of the finest enhancements available today for the Apple][computer. It increases the utility and potential for the machine, and frankly, is a pleasure to use. For the software experimenter, it is like having one (or maybe two) new machine(s), all for the cost of a few trips to the grocery store.

The Z-80 CP/M system does not see the world in the same way as does the 6502-based computer. Accordingly, the Apple's memory and disk maps are reorganized to reflect the new view of things. The Apple Z-80 memory map has been thoroughly described in a recent article by Wes Huntress. In the present article, I will try to elucidate the Apple CP/M disk map. CP/M (The Softcard's operating system) comes in 13-sector and 16-sector versions, but I will only discuss the 16-sector version-and I am assuming that you already know a little about DOS and CP/M.

Most disk and memory references are in hexadecimal, with "$" or "H" added for clarity. Please remember that 6502 addresses are different from Z-80 addresses. The Z-80 would see the 6502 address, $0300, as 0F300H.

A disk must be formatted before it can be used. The CP/M formatter creates the usual 35 tracks of 16 sectors, but sets every data-byte on the disk to $E5. The $E5 value serves a useful purpose only in the directory portion of the disk to indicate an erased file.

The newly formatted disk is quite ordinary, and can be dumped with any of the usual 6502 based disk utility programs. It can also be dumped from within CP/M, but that involves notions beyond the focus of this article.

No file, directory, or system code is placed on the disk by the FORMAT command. Many eight inch disks come factory formatted with $E5. Consequently, they are ready for immediate use on CP/M systems.

The formatted disk will accept data, but will not boot until it has an operating system. CP/M is put onto the disk with the COPY command, and ends up on tracks $0, $1, and $2—just where you would expect.

The operating system is written mostly for the Z-80/8080, but there is a good amount of 6502 code on track $0. This is reasonable, since the Z-80 needs considerable support from the 6502.

A 6502 I/O package is found on track $0, beginning at sector $9. It includes a modified (different from DOS 3.3) RWTS, with its associated read, write, nybble, format, set-up, and miscellaneous routines. The RWTS boots to memory location 0FE00H, with the other routines loading just below it. You might enjoy trying to decipher the I/O control block (IOB), which begins at 0F3E0H.

CP/M considers files to be comprised of 128-byte RECORDs. A large file may consist of several EXTENTs of 128 records each. Such an organization conflicts with the way in which data must be stored on the disk (256-byte sectors), but the BIOS portion of CP/M automatically combines records (to fill up sectors) if necessary.

Disk file space is allocated in 1024-byte "blocks." Therefore, any file created under CP/M, even a file of only a few bytes, occupies at least four sectors. This allocation scheme may appear a bit inefficient at first, but it tends to keep files on as few tracks as possible, and so reduces access time—CP/M's disk access is very fast.

Is your abacus handy? There are four blocks per track, and each block is numbered. Blocks $0, $1, $2, and $3 are on track $3, and the numbering continues on track $4, up to the last track on the disk.

There are 128 blocks available to the system. You can find the track-location of any block by starting at track $3, and counting, four blocks per track, until you get to the block-number you want. Or, if you prefer, track-number = int (block-number / 4) + 3.

Now for the best part! The sectors within each block are not contiguous, but are scattered about the track to increase speed. The ordering of sectors within a block follows one of four patterns:

Pattern-number	Pattern
$0	$0, $6, $C, $3
$1	$9, $F, $E, $5
$2	$B, $2, $8, $7
$3	$D, $4, $A, $1

The pattern-number for a block is its block-number modulo 4.Pick a block, say block $9. Well, 9 modulo 4 = 1, so block $9 (on track $5) follows pattern $1. Suppose that the system is to store five records in block $9. CP/M allocates all four sectors of the block, and stores two records in sector $9, two more records in $F, and a final record in sector $E. Sector $5 is allocated, but unused. Blocks $0 and $1 are reserved for the directory.

Let's consider one more example before moving ahead. Suppose that you send eleven pages of the Apple's memory to a freshly formatted disk. CP/M allocates the first three available blocks: $2, $3 and $4. The file is threaded onto tracks $3 and $4 in the following order: track $3, sectors $B, $2, $8, $7, $D, $4, $A, $1; track $4, sectors $0, $6, $C. The file consists of 22 records, and occupies 11 sectors in three blocks. Sector $3 on track $4 is allocated, but unused.

Once Apple-C/M disk allocation and blocking are understood, the organization of the directory is easily grasped. It can hold up to 64 entries, however, for some reason Microsoft limited the number to 48. Each entry is 32 bytes long, and may be partitioned into seven fields,

```
u ffffffff ttt e ss r
aaaaaaaaaaaaaaaa,
```

where

u is the user-number
f is the primary filename
t is the type or secondary filename
e is the extent-number
s is for system use in the file control block
r is the number of records,and
a is the allocation for the file

```
]LIST
10   REM SO WHAT DID
       YOU EXPECT
JAY  BAKST :CALL -A.P.P.L.E.

20   DIM A(9,2)
30   HOME : PRINT "SO
       WHAT DO YOU EXPECT?";
40   FOR PAUSE = 1 TO
       500: NEXT
50   FOR X =  - 8 TO
       8 STEP 2
60  X1 = X / 2 + 5
70  Y =  INT ( SQR (100
       - X * X) + 1.5)
80  A(X1,1) = Y:A(X1,2)
       = 13 - Y
90   NEXT X
100  SO$ =  CHR$ (( PEEK
       (36) - 2) * 4 + 3)
110  WHT$ =  CHR$ ((
       PEEK (36) - 3)
       * 4 + 3)
120   HOME : SPEED= 50
130   FOR I = 1 TO 9
140   VTAB (I + 1): HTAB
       (A(I,2)): PRINT SO$
150   VTAB (I + 9): HTAB
       (A(I,1)): PRINT SO$
160   IF I < 5 THEN
       VTAB (I + 1):
       HTAB (A(I,1)):
       PRINT SO$
170   IF I > 5 THEN
       VTAB (I + 9):
       HTAB (A(I,2)):
       PRINT SO$
180  Z = 2 -  INT ((ABS
       (5 - I) + 1) / 2)
190   VTAB (10 - I) *
       2: HTAB (A(10
       - I,2)) + 20 -
       Z: PRINT WHT$;
200   VTAB (I) * 2: HTAB
       (A(I,1)) + 21
       + Z: PRINT WHT$;
210   NEXT I
220   SPEED= 255
230  I =  PEEK (37) +
       2:J =  PEEK (36)
       - 22: POKE 37,I:
       PRINT : POKE 36,J
240   PRINT "WHAT DID
       YOU EXPECT?"
```


Improve Your Programs on the APPLE II

Your favorite routines belong with the most efficient "power-boost" possible for the II/II+. **THE ASSEMBLER DEVELOPMENT SYSTEM** provides the cleanest path to interface the powerful 6809 with DOS.

STELLATION TWO

The Lobero Building P.O. Box 2342
Santa Barbara, Ca. 93120
(805) 966-1140 Telex 658439

DISK MAPPING THE Z-80

An example or two may be helpful. I have separated the following directory dumps into fields, although they would not appear that way on the disk.

The user-prefix indicates that this file belongs to user $0. The filename is entirely in low-ASCII, which is the usual state of affairs, however either the system or the user may set certain bytes high to give the file new attributes. Do this carefully; your file could easily obtain unwanted or meaningless attributes. The type-qualifier for this file is also in low-ASCII. This entry controls extent $0 of the file, and since the file is small, there are no other extents. The two bytes of the s-field are perplexing. I have not found them to be active in the directory, and suspect that their main use is in the control block which the system sets up in memory for each open file. The present file consists of two records (one sector) and is stored in block $2. Below is another directory entry, $2.

A close look at the entry shows that the file belongs to user $3. Its 10 records are stored in blocks $3 and $4. Notice that the second byte of the t-field is in high-ASCII, an indications that the file has the SYS (system) attribute.

When a file is erased, its user-number is set to $E5. You can restore the file (if its allocation is intact) by setting its user-number to something more reasonable, such as $0.

A very large file may require multiple directory entries as shown in entry #3.

The file is 33K in length! It must be partitioned into three extents; the e-fields confirm that the entries control extents $0, $1, and $2. The first byte of each t-field is high, giving the file the R/O (read/only) attribute. Extent $0 contains 128 records, and is written into blocks $5 through $14. Extent $1, also with 128 records, is found in blocks $15 through $24. But extent $2 has only four records, which fit nicely into block $25.

There it is. Initially, it looks complicated, but upon further reflection, it becomes elegant and simple. You have probably noticed that the organization of the directory seems to obviate the need for separate track-bit maps and track-sector lists.

As you have also probably discerned, I am very enthusiastic about Apple CP/M. Much of the information for this article was gathered the hard way—by dumping and analyzing CP/M disks, sector by sector. If you find any mistakes, please let me know. If you should uncover something new about Apple CP/M, please let us know that too.

The *CP/M Alteration Guide* (not supplied with the Softcard) would be useful for those wanting to know more about the BIOS module of CP/M, which determines how CP/M interfaces to specific hardware (e.g., Apple 16 or 13 sector disks). This manual may be ordered from Digital Research.

Entry 1

```
00  40494E4946494C45  484558  00  0000  02  0200000000000000...00
    ↑↑                ↑       ↑   ↑     ↑   ↑
    u f               t       e   s     r   a
```

Entry 2

```
03  4049444946494C45  48c558  00  0000  0A  0304000000000000...00
    ↑↑                ↑       ↑   ↑     ↑   ↑
    u f               t       e   s     r   a
```

Entry 3

```
00  4D45474146494C45  C84558  00  0000  80  05060708090A0B0C...14
00  4D45474146494C45  C84558  01  0000  80  15161718191A1B1C...24
00  4D45474146494C45  C84558  02  0000  04  2500000000000000...00
    ↑↑                ↑       ↑   ↑     ↑   ↑
    u f               t       e   s     r   a
```

– MORE –
ON
CRACK-SHOT

- ★ CRACK-SHOT is easy to use
  Designed for Amateur and Professional
- ★ Automatic B-file maker
  B-file can run on another APPLE
  without the card installed.
  B-file may be copied COPY A
  B-file may be sent over Modems
- ★ Fantastic Utilities
- ★ Huge Documentation
- ★ HOT-LINE 24 hours/day 7 days/week
  for any system upgrades thru
  PIRATES HARBOR
  (Over 1000 Users Now) $15.00 Extra
- ★ Great Product Support
- ★ CRACK-SHOT hardware contains 8 chips & 2 switches including E-prom and add on memory. Use in any slot other than 6.

For Mastercard or Visa Only

Call Now! TOLL FREE

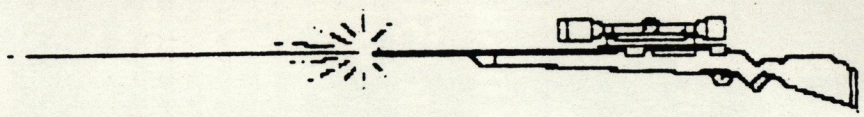
1-800-824-7888

Ask for Operator 68

in CA 1-800-852-7777

Ask for Operator 68

CRACK-SHOT



THE ULTIMATE IN COPY DE-PROTECTION

Not just another bit copier.

Blows Locksmith and Nibbles Away — AWAY

Even cracks all bit copiers in fifteen seconds

CRACK-SHOT is a hardware device that fits right in an Apple Slot and dumps all memory to disk.

- ★ Use as a gaming tool to stop, start, and save a game at any level.
- ★ Full monitor capabilities to examine, modify, trace single-step, or disassemble an interrupted program
- ★ Copies from DOS 3.2 to DOS 3.3
- ★ Faster and easier to use than any bit copier
- ★ Lets you suspend work with one program while you use another
- ★ Cracks single disk access programs in fifteen seconds, longer for multi-access

System Requirements: 48K Apple II or Apple II Plus
1 Disk Drive
Ramcard helpful but not required

Complete with Software and Instructions. $149.95

Cash, Check, Money Order, Mastercard of Visa accepted

Order now from:

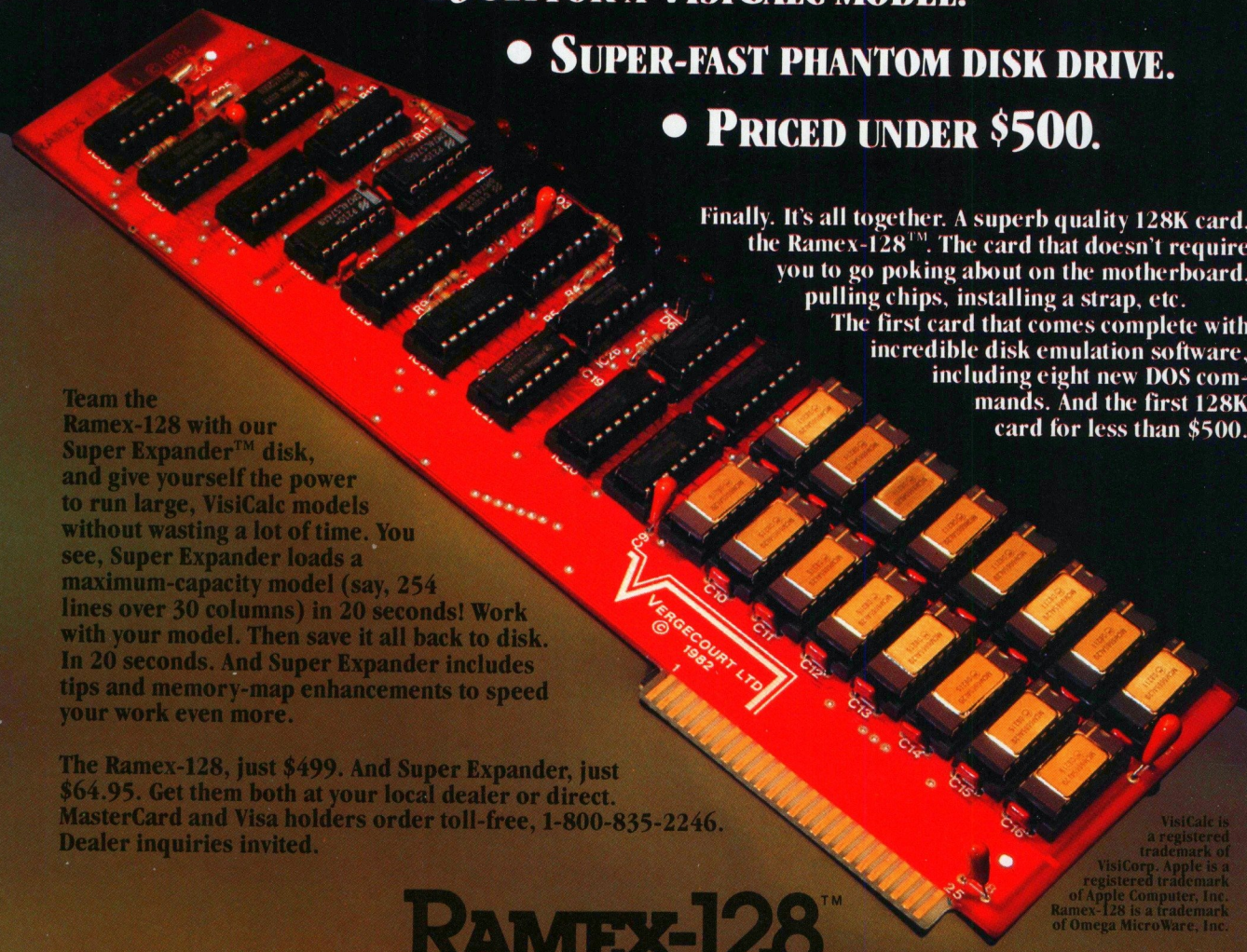
PIRATES HARBOR

P.O. Box 8928, Boston, MA 02114 617/738-5051 MODEM

Apple is a registered trademark of Apple Computer, Pirates Harbor is the Trademark of the Red Rebel Locksmith—trademark of Omega Microware, Inc. Nibbles Away—trademark of Computer: applications

NOW HERE FOR YOUR APPLE®: A CARD THAT BLOWS ALL THE OTHERS AWAY!

- LOADS OR SAVES A FULL VISICALC® MODEL IN 20 SECONDS. (NOT 16 MINUTES LIKE THE OTHERS)
- 136K FOR A VISICALC MODEL.
- SUPER-FAST PHANTOM DISK DRIVE.
- PRICED UNDER $500.



Finally. It's all together. A superb quality 128K card, the Ramex-128™. The card that doesn't require you to go poking about on the motherboard, pulling chips, installing a strap, etc.
The first card that comes complete with incredible disk emulation software, including eight new DOS commands. And the first 128K card for less than $500.

Team the Ramex-128 with our Super Expander™ disk, and give yourself the power to run large, VisiCalc models without wasting a lot of time. You see, Super Expander loads a maximum-capacity model (say, 254 lines over 30 columns) in 20 seconds! Work with your model. Then save it all back to disk. In 20 seconds. And Super Expander includes tips and memory-map enhancements to speed your work even more.

The Ramex-128, just $499. And Super Expander, just $64.95. Get them both at your local dealer or direct. MasterCard and Visa holders order toll-free, 1-800-835-2246. Dealer inquiries invited.

VisiCalc is a registered trademark of VisiCorp. Apple is a registered trademark of Apple Computer, Inc. Ramex-128 is a trademark of Omega MicroWare, Inc.

RAMEX-128™

UNDER $500. WHAT ARE YOU WAITING FOR?

OMEGA MICROWARE, INC.™

222 SO. RIVERSIDE PLAZA • CHICAGO, IL 60606 • 312-648-4844

PROGRAMMING AND PROBLEM SOLVING

Dr. David A. Lingwood

THIS article began as a letter to the *A.P.P.L.E. Doctor*, in answer to a question in the June *Call -A.P.P.L.E.* about precision — or the lack of it — in mathematical and statistical calculations (June, p. 77). As the letter "wrote itself," it became obvious that there is a broader lesson about finding and using *information resources*, or perhaps better yet, about *problem solving methods*, that is more important than the solution of the particular problem.

So, this article is divided into two parts. I will talk about the specific numerical accuracy problem first, then go on to the more general resources problem, using accuracy as a specific example.

The Accuracy of Statistical Calculations

The *A.P.P.L.E. Doctor* is generally correct that "answers were only as accurate as my least accurate measurement." However, single precision *may* be sufficient in statistical summations, *or* may cause severe rounding error problems depending on the magnitude of the original data.

Most of the problems the reader mentions occur through blind use of what are known as "machine calculation formulae" for things like means and sums of squares. "Machine" here refers to calculators, not computers. These formulae were developed in the early days when all researchers had available were large electric calculators.

As an example, let's look at the "definitional" and "machine" formulae for statistical variance. Any statistics book will give these as:

(definitional)

$\Sigma x^2/N$

where x is the deviation of a single data value from the mean of all values

This implies you need to know the mean in advance — i.e., that you enter all data values first to find the mean, then enter all the deviations in a second pass — computationally inefficient. Hence the "machine" formula:

(machine) $N \Sigma X^2 - (\Sigma X)^2$

The sigma (Σ) means "sum of." This lets you make one pass through the data. To compute a mean and a variance, the researcher would enter data one value at a time, and the calculator would store three sums: 1) the number of entries (i.e., "N"); 2) the sum of the data values (sum X); 3) the sum of squared data values (sum X-squared). These values are then plugged into the formula.

If data values are small relative to the storage accuracy of the machine, everything is fine, and the results are accurate. However, if the data values are large to begin with, and/or there are many such values, there can be substantial rounding error in storing the sum of the *squared* data values. Goodnight (1979, p. 151) says, "Any program written in single precision is obviously in trouble when more than four significant digits appear in the input data." Kennedy and Gentile (1980, p. 320) note, "for a given floating-point level of precision, there will always be data sets that are sufficiently ill-conditioned so that they cannot be processed accurately."

The definitional formulae are more accurate since there is a subtraction of the mean from the data values before taking the sum. So, the values summed are smaller.

The problem is even worse in calculating correlations between *two* measures or variables. Here one needs to store the sum of the squared *products* of pairs of data values.

What to do? Either use multiple precision, or change formulae. Applesoft, FORTRAN, etc. lack double precision, so one obviously needs different formulae.

The simplest approach is to go back to the definitional formulae, making two passes through the data. Calculate the sum of cases (N) and the mean in the first pass, then the sum of deviations from the mean. If you don't have so many observations that calculation time becomes a factor, this is fine. In fact, I would always recommend this approach for the student who wants to learn how various statistics work by writing programs for them. However, in a "production" statistics package, time becomes very important.

There is a solution, however, generically known as "provisional means" algorithms. The strategy here is conceptually simple. The formula you use calculates, at each observation, the difference between the present data value and the results of the definitional formula based on all the previous observations, then adds this difference in to the total. In the case of means and variances, at the end of one loop through the data, the two storage variables used will contain the mean and the sum of squared deviations from the mean ("little x squared") directly.

Kennedy and Gentile (p. 323) give formulae for means and the cross products needed in a correlation. An Applesoft loop for means and variances looks like this:

```
N = 0: MN = 0: SX = 0: REM # CASES, MEAN, AND SUM OF SQ. DEVIATIONS
FOR I = 1 TO NC: REM STEP THROUGH NC OBSERVATIONS OR "CASES"
    N = N + D(I) : REM D( ) IS VECTOR OF DATA VALUES
    X = D(I) - MN : REM DIFFERENCE BETWEEN DATA AND PREVIOUS MEAN
    MN = MN + X / N : REM UPDATE PREVIOUS MEAN VALUE
    SX = SX + X * (V - MN) : REM UPDATE PREVIOUS SUM SQ. DEVIATIONS
    NEXT I
PRINT "N = ";N;" MEAN = "MN
PRINT "VARIANCE = ";SX/(N-1)
```

How accurate is it? A user of my AIDA statistical analysis package on the Apple] [has recently done comparisons of the AIDA correlation routine against two mainframe statistical packages (OSIRIS and MIDAS), which use *double* precision machine formulae on an Amdahl 470. AIDA's provisional means results give *greater* accuracy!

Information and Problem Solving

Information Problems

The reader is not alone in struggling with accuracy problems, and this is the most important thing for him/her to realize. Computer science *is* a science. This means two things:

1. Knowledge is *cumulative*: there is a large literature of research and algorithms dealing with any problem a microcomputer programmer is ever likely to encounter. This literature builds on — and references — earlier knowledge.
2. Knowledge is *tested*: solutions to problems are verified and compared. Dimensions of comparison usually involve either the accuracy and/or efficiency (i.e. speed) of computation.

Only a part of the "science" has rubbed off on the microcomputer literature. Authors and editors need to be taken to task for:

- not testing the programs they publish
- not insisting that authors reference previous articles
- not comparing one program with similar ones.

Moreover, published reviews will only rarely test against standard benchmarks, and even then, the accuracy of computation is infrequently tested with rigor. Why? Because testing for accuracy requires running a program with a full range of possible data, looking for obscure bugs or errors that may pop up only in certain rare circumstances. In short, it is darned hard work.

Indeed, about all the comparison that ever gets done is on *speed*. Few articles ever test for *accuracy*, though in many cases accuracy is (we hope!) self-evident. Some of the scientific tradition is evident in development of "benchmarks" through which computers and languages are compared, but even here, too many authors develop their own benchmark programs, rather than researching what benchmarks already exist in the literature. If the author uses his/her own benchmarks, then the knowledge gained is very limited by being non-cumulative.

Why do we have these problems? Simply, many authors and editors are not scientifically trained. In fact, microcomputer "science" may be the first instance of a field in which scientific or technical knowledge emerged from a field not dominated by scientists themselves. Put another way, microcomputing is the only field I know of (and my own research field is technology transfer) in which knowledge producers and users mingle equally.

Microcomputing is thus very "action oriented." Users-cum-authors often sit down and whip out programs to solve immediate problems without reference to anything except the Apple manual. No research. No complete definition of the problem itself.

Micros encourage this kind of spontaneity: their ease of use, and the ease with which programs can be changed, make it simpler to just plunge in than to define and research the problem first.

This problem has a brighter side, however: only the microcomputer literature has added the important third aspect of a user-oriented science:

3. Knowledge should be *useful*: the microcomputer literature tests for and reviews "friendliness" of programs quite well, while the traditional literature in computer science completely ignores the issue of how easy or hard programs are to use. Hooray for us!

The very growth of our literature and available programs causes problems, however. Indices of commercial programs exist, and there is an index to the literature. Still, the field is growing so fast that it is very hard to keep current.

What does this mean for the reader with a problem? First, you should be aware that articles in the microcomputer literature will only rarely reference other articles — either in that same literature or (a more serious lack) in the general literature of computer science. There may be more efficient articles in other sources. The article you read now may be, in effect, a "plagarism" of some other author's work, and if it lacks references you may never know. Beyond this, articles themselves can't keep up with the growth of the literature.

Problem Solving Steps

We need to back up a bit, however, before charging off into either the literature or programming itself. I'd like to suggest some steps for "problem solving," as the process applies to the writing or fixing of micro programs. I'll use the example from the statistical accuracy issue.

1. Know You Have a Problem

The first problem is to figure out whether or not you *have* a problem. The reader said he/she couldn't get calculations to come out right. Obviously, a program had been written that didn't compare favorably with known correct answers. In short, the reader had done some testing, and had accurate standards against which to work. Where do you find standards? In statistics and numerical analysis there are two kinds: 1) textbook examples, and 2) results from assumedly-accurate commercial "mainframe" packages.

The *Collected Algorithms of the ACM* contain many examples with data and results (see below). This is an excellent place to start if your library doesn't contain any examples.

Testing should be done over the complete range of possible data values. In the case of the variance example, above, you would never see the effects of rounding error if you used only a few numerically-small data values in testing (and many of the textbooks use only such simple data — so beware).

2. Look for Resources

In many cases, though, the "problem" itself may be that you don't know how to write the program to do whatever you want. In either case (don't know, or looking for test data), you need to move into an "information search" mode.

This is a good place to introduce a valuable information resource: the *Collected Algorithms of the ACM* (Association for Computing Machinery). Not only does this collection cover most of the numerical, sorting, statistical, etc. algorithms in existance, but it contains certification

tests with real data and results. The algorithms are written either in FORTRAN or a generic structured language that looks much like ALGOL or Pascal. Let's look at one example. It is algorithm 66, by J. Caffrey, a simple matrix inversion method:

```
procedure Invrs (t) size : (n); value n;
real array t, integer s
begin integer i, j, s; real array
v[1:n-1]; real y, pivot
for s := 0 step 1 until n-1 do
   begin pivot =: 1.0 / t[1,1]
      for i := 2 step 1 until n do
      v[i-1] := t[1,i]
      for i :- 1 step 1 until n-1 do
         begin t[i,n] := y := -v[i] *
         pivot
         for j := i step 1 until n-1
         do
         t[i,j] := t[i + 1,j + 1] + v[j] *
         y
         end
      t[n,n] := -pivot
   end
   for i := 1 step 1 until n do
   for j := i step 1 until n do t[i,j]
   := -t[i,j]
end Invrs
```

(the only change I've made is to substitute "*" for the multiplication sign, to avoid confusion with the letter "x"). Even if you don't know the language, it is easy to rewrite this into Applesoft or whatever. More important, there are certifications, with test matrices and results, against which a rewrite may be compared. One of the certifications suggests an improvement (a "singular matrix" error exit if pivot = 0 before the first *i* loop).

Something else is worth noting. The original algorithm appeared in *Communications of the ACM* in **1961**! This algorithm is old enough to vote. Time implies that the algorithm has passed the test of time. Also, many of the algorithms developed in the "old days" ran on machines that were limited in memory and/or precision. Thus, they work well in the Apple environment.

There are also many valuable reference books on computing. Subfields such as statistics, numerical analysis, sorting, databases, etc. have their own references. Where do you find them? The best source (as for the *Collected Algorithms*) is the library of a good university computer science department. Even if you're not a student, you can usually use such a library's materials on site. Begin by telling the librarian your problem.

If you want to build your own library, you could buy a professional computer science publication, to pick up ads for computer science books. Exhaust the library resources first, however — both to learn more about the topic, and to learn which books you need.

Scientists always begin solving problems this way, right? Well, *no*. The myth is that scientific problem formulation and information gathering is based on the published literature. You might first do what scientists *actually* do: they talk to or call somebody. Every scientific field contains experts known by their peers to be "up" on the literature, and on who is doing what. These folks field a lot of calls, being paid off for their trouble by thus learning even more. The most productive scientists are those who are best plugged into the informal communication networks.

How do you plug in? First, *A.P.P.L.E.* has its own experts — our consultants and our hotline. In more specific or technical problems, check in your company or through a professional association's local chapter. Even if you live in East Nowhere you can usually find an "old hand" computer scientist with back issues of ACM publications, etc.

If all else fails, find the most recent author of a technical article in the field, an article with good references, then *call* him or her. Be prepared first, though: ask the right questions. You may find that you need two kinds of experts: one for the technical problem, and one for advanced uses of the Apple.

3. Organize

There is a logic to programming that is higher than the simple rules for writing programs in any given language. Texts such as Nagin and Legard (1978) teach "top down," "structured" programming methods. The point is to design in modular sets of routines, doing the design at a conceptual level first. You can no more learn to program, say, Applesoft, from the Applesoft manual than you can learn to write by reading a dictionary.

The key design element is that of being certain what is *causing* your problem before you design the solution. In the statistical example, the problem was not really lack of precision, but rather, choice of algorithm.

4. Code

This is where people usually begin learning to program. It is much too far a point downstream to begin. The manuals, magazines, classes, etc. all cover this level well, so I won't.

5. Test and Evaluate

One reason for modular programming is so that you can test each routine by itself. A good idea.

You can't really test a program yourself, either, any more than a writer can be their own editor. Try out your program on friends, club members, etc. If you have young children, have them try your menus and read your documentation. If a five year old can do it, there is some (perhaps faint) hope that real-world users can make the program work and not crash it.

6. Implement

Someone once said that the object lesson for revolutionaries is having to run a government. The object lesson for programmers is having users.

SUPER CALCULATOR

Turn your APPLE II into a powerful programmable calculator.

- 9 Single keystoke user defined functions
- 24 basic arithmetic functions
- Special memory operations
- Immediate calculator mode
- Program mode
- Powerful calculator based program language
- Printer interface
- Built-in functions
- Program Editor

REQUIRES APPLE II WITH DISK DRIVE

$24.95

SRA

P.O. Box 44018

Tacoma, WA 98444

Washington residents add 6.6% sales tax

A.P.P.L.E. Members Only

Introducing:

AP•Jewelry

This ½x¾" Gold on Red Lapel Pin is the hit of computer shows, meetings or even work.

Show pride in being part of the **Number One** Apple User Group in the world.

Member Price: **$3.00**

A.P.P.L.E. Orders
304 Main Ave. S., Suite 300
Renton, WA 98055
(206) 271-4514

Washington State residents add 7.0% sales tax.
Overseas shipping add $6.00
Canadian rates being revised

"Production" program users will find unimagined bugs and traps in even the best code? Why? Because the author and other sophisticated users know too much about what the program is supposed to do and how. They don't make the simple errors. ("Oh, you mean I have to press return?").

7. Contribute to Our Science

Ultimately, you as a programmer can have a maximum inpact (as well as pay back your debts) only if you contribute to the literature itself. Writing also helps you understand what you are doing much better. If you are a commercial programmer, you need not give away the proprietary store. There are always useful techniques and tips through which others may benefit.

Finally

The ultimate lesson from this is not so much the solution to a particular programming problem, as it is a plea for Apple users to treat what they do in a more scientific manner: engage in a cumulative, problem-solving approach to their work. There is really no excuse for inaccurate statistics, abominably slow sort algorithms, reference-less articles, etc. Microcomputerists will never be taken seriously until they learn about and build from a solid previous base, and in turn, contribute to it.

REFERENCES

Caffrey, J., Algorithm 66: INVRS. Collected Algorithms from CACM. Association for Computing Machinery. Originally published in Communications of the ACM, July, 1961.

Goodnight, J. H. (1979), A Tutorial on the SWEEP Operator, The American Statistician, August, 149–158.

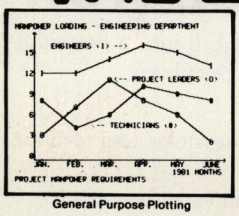
Kennedy, W.J., Jr., and Gentile, J.E. (1980), Statistical Computing. New York: Marcel Dekker.

Nagin, P.A., and Legard, H.F. (1978), BASIC With Style. Rochelle Park, N.J.: Hayden.

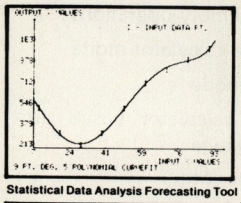
THE SUPERPLOTTER©

version 2.2

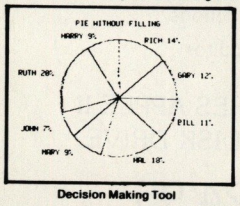
For use with Apple™ Computer Systems



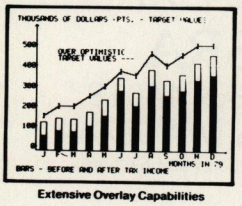
General Purpose Plotting



Statistical Data Analysis Forecasting Tool



Decision Making Tool



Extensive Overlay Capabilities

The Superplotter is a highly versatile business, engineering, educational, math and graphics applications package featuring:

Standard Bar Graphs•Point and Line Graphs•Graphics Display of any Mathematical Function•Least Squares Polynomial Curvefit Generation•Keyboard Image Shape Tables and User Tutorial•Automatic Graphics Disk Storage and Recall•Data File Editor•Automatic Disk Storage and Recall of Editor Data Files•Overlay Modes•Graphics Screen Text Editor

Data may be input directly or from previously generated data files. Comprehensive documentation is provided including printer dump instructions and directions for accessing data bases created by other programs. Requires Applesoft and one DOS 3.3 disk drive.

$69.95

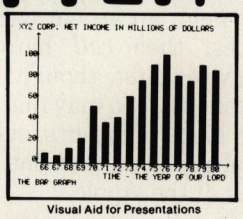
See your local dealer for a demonstration or order direct from:

Dickens Data Systems

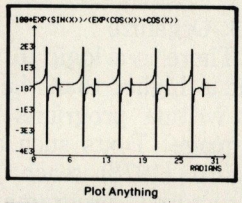
478 Engle Dr. / Tucker, Ga. 30084 / (404) 491-7905

outside Georgia:
(800) 241-6753 ext. 503

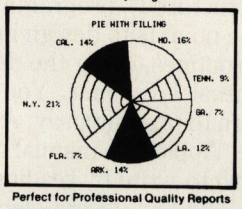
C.O.D., VISA, & MasterCard accepted.
Mail Orders: Add $2.50 for shipping and handling ($5.00 outside the USA)



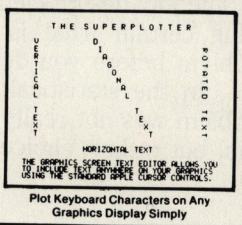
Visual Aid for Presentations



Plot Anything



Perfect for Professional Quality Reports



Plot Keyboard Characters on Any Graphics Display Simply

A.P.P.L.E. Members Only

AP. COOLKIT

Now Available
A.P.P.L.E. Members

$22.50

Consisting of a Torin slow speed fan, line cord and mounting tape. (See Keep Your Cool, Call — A.P.P.L.E., May 1982)

A.P.P.L.E Orders
304 Main Ave. S., Suite 300
Renton, WA 98055
(206) 271-4514

Washington State Residents add 7.0% sales tax

Sorry, not available to foreign or overseas members.

Shipping Wt. 16 oz.

NOW AVAILABLE

A.P.P.L.E. Member Price: $11.00

Non-Member Price: $12.50

(Available at your local computer store)

184 page reference tool containing a wide range of new and useful articles/programs specializing in Apple Pascal.

A.P.P.L.E. Members Only
Finalize Your Pascal Needs!

IN DEPTH II DISKETTES

Complete set of three diskettes containing all programs listed in In Depth II.

Member Price:

$20.00

Foreign shipping charge: $12.50
Washington state residents add 7.0% sales tax.

A.P.P.L.E. ORDERS
304 Main Ave. S., Suite 300
Renton, WA 98055

Foreign members add $6.00 shipping/handling charge.

Canadian members, postage being revised.

Dealer inquiries invited.

Washington state residents add 7.0% sales tax.

SHOPPING LIST

- personal database management program for construction of shopping lists and handling of similar information
- speeds shopping list preparation, minimizes forgotten items
- accommodates up to 500 user-definable items per file, multiple files can be used
- supplied with a database of over 200 common grocery-type items already on disk
- print shopping lists sorted by category for convenient use

CALENDAR

- personal database management program for date-related information
- remember appointments and important dates, log maintenance records, etc.
- automatic generation of on-screen or printed calanders for any month in this century

MONEY MANAGER

- professional quality home accounting system
- handles checkbook, cash, charge card, savings account and virtually any other type of transaction
- automatic and coded input features minimize keystrokes and speeds data entry
- interactive budget creation and automatic budget tracking - budget status can be checked at any time, even in the middle of a month
- complete series of on-screen and printed reports
- lightning fast checkbook reconciliation
- check printing
- multiple file audit capabilities

INTRODUCING THE OMNIWARE PERSONAL SERIES

PERSONAL PACK

- includes Personal Money Manager, Personal Calendar, and Personal Shopping List all on one diskette

OMNIware

- **Professional quality • Unlocked diskettes • Listable code**
- **Requires Apple II with Applesoft in ROM, 48K RAM, DOS 3.3**

Distributed by
Educational Computing Systems
106 Fairbanks Rd., Oak Ridge, TN 37830
615-483-4915

PERSONAL SHOPPING LIST - $24.95
PERSONAL CALENDAR - $24.95
PERSONAL MONEY MANAGER - $59.95
PERSONAL PACK - $89.95
OMNITEST - $34.95
Add $2 shipping, in TN add 4½% tax
VISA and MASTERCARD Accepted

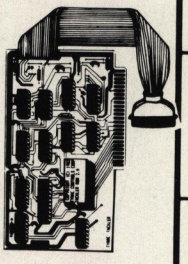
OMNI is a trademark of ENDAC, Inc.

OMNITEST

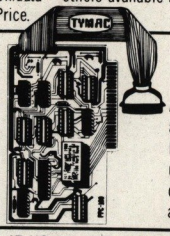
- flexible educational system for home, school, or anywhere learning is important
- authoring system allows you to design and construct your own quizzes and drills on any subject, at any level
- multiple choice, fill-in-the-blank, true-false questions allowed
- includes interactive drill program and challenging quiz game
- includes vocabulary, state capitals, and world capitals quizzes already on disk

APPLE HARDWARE

THE TACKLER ™ — DUAL • MODE PARALLEL INTERFACE FOR THE APPLE® 2 BOARDS IN ONE FOR NO MORE COMPATIBILITY PROBLEMS!



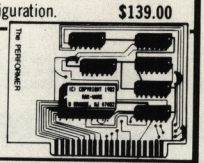
An intelligent board to provide easy control of your printer's full potential. Plus a standard parallel board at the flip of a switch – your assurance of compatibility with essentially all software for the APPLE®. Hires printing with simple keyboard commands that replace hard to use software routines. No disks to load. Special features include inverse, doubled, and rotated graphics and many text control features, available through easy keyboard or software commands. Uses Industry standard graphics commands. This is the first truly universal intelligent parallel interface! Change printers – no need to buy another board. Just plug in one of our ROM'S and you're all set. ROM'S available for Epson, C. Itoh, NEC, and Okidata – others available soon. Specify printer when ordering. Call for Price.



THE UPGRADEABLE PPC-100 PARALLEL PRINTER CARD

A Universal Centronics type parallel printer board complete with cable and connector. This unique board allows you to turn on and off the high bit so that you can access additional features in many printers. Easily upgradeable to a fully intelligent printer board with graphics and text dumps. Use with EPSON, C. ITOH, ANADEX, STAR-WRITER, NEC, OKI and others with standard Centronics configuration. **$139.00**

IF YOU WANT GRAPHICS AND FORMATTING THEN CHOOSE THE PERFORMER



for Epson, OKI, NEC 8023, C. ITOH 8510 provides resident HIRES screen dump and print formatting in firmware. Plugs into Apple slot and easy access to all printer fonts through menu with PR# command. Use with standard printer cards to add intelligence. **$49.00** specify printer.

THE MIRROR FIRMWARE FOR NOVATION APPLE CAT II®

The Data Communication Handler ROM Emulates syntax of an other popular Apple Modem product with improvements. Plugs directly on Apple CAT II Board. Supports Videx and Smarterm 80 column cards, touch tone and rotary dial, remote terminal, voice toggle, easy printer access and much more.
List $39.00 Introductory Price **$29.00**

MINI ROM BOARDS

Place your 2K program on our Mini Rom Board. Room for one 2716 EPROM. Use in any slot but zero. Only **$34.95**

DOUBLE DOS Plus

A piggy-back board that plugs into the disk-controller card so that you can switch select between DOS 3.2 and DOS 3.3 DOUBLE DOS Plus requires APPLE DOS ROMS. **$39.00**

APPLE SOFTWARE

Super Pix

Hires screendump software for the Epson, OKI, C. Itoh and Nec 8023. Use with Tymac PPC-100. Special **$19.95** (Specify Printer)

Mr. Lister – Customer Contact Profiler & Mailer

A Super Mail List Plus more — up to 1000 Entries on single 3.3 Disk (only 1 Drive required) — 2 second access time to any name — full sort capabilities — Dual Index Modes — supports new 9 digit Zip. Easy to follow manual — Not Copy Protected — 4 user defined tables with 26 sort selections per table — Beta tested for 6 months — user defined label generation.
Introductory Price $135. **$99.00** Dealer &: Dist. Inquiries Invited.

APPLE LINK

A communications system for the Apple® (Requires Hayes Micro Modem). Transmit and receive any type of file between APPLES®, Automatic multi-file transfer, real time clock indicating file transfer time. Complete error check. Plus conversation mode. Only one package needed for full transfers. Compatable with all DOS file types. (requires Hayes Micro Modem) **$59.00**

THE APPLE CARD/ATARI CARD

Two sided 100% plastic reference card Loaded with information of interest to all Apple and Atari owners. **$3.98**

NIBBLES AWAY II

AGAIN! Ahead of all others.

- **AUTO-LOAD PARAMETERS . . .** Free's the user from having to Manually Key in Param values used with the more popular software packages available for the Apple II.
- **EXPANDED USER MANUAL . . .** incorporates new Tutorials for all levels of expertice; Beginners Flowchart for 'where do I begin' to 'Advanced Disk Analysis' is included.
- **TRACK/SECTOR EDITOR . . .** An all new Track/Sector Editor, including the following features: Read, Write, Insert, Delete Search, and impressive Print capabilities!
- **DISK DIAGNOSTICS . .** Checks such things as: Drive Speed, Diskette Media Reliability, and Erasing Diskettes.
- **HIGHEST RATED . . .** Best back up Program in Softalk Poll (Rated 8.25 out of 10).
- **CONTINUAL UPDATES . . .** Available from Computer Applications and new listings on the source. **$69.95**

Dealer and Distributor Inquiries Invited.

VISA MasterCard

MICRO-WARE DIST. INC.
P.O. BOX 113 POMPTON PLAINS, N.J. 07444
201-838-9027

EXTRA, EXTRA READ ALL ABOUT IT

Craig Peterson

NOW that 16K RAMcards for the Apple][are dropping to disgustingly low price levels, I'm sure that many of *Call – A.P.P.L.E.'s* readers are planning to or have already purchased one. Besides the obvious advantages of having both Applesoft and Integer Basic available on your one little computer, you can also obtain a DOS moving program to place DOS on the RAMcard. This really gives Applesoft a lot of real estate to play with. Cornelis Bonger's DOS MOVER which appeared in the July/August 1981 issue of *Call – A.P.P.L.E.* is an excellent one, and with some keyboard action on your part, all it costs is the price of a back issue. However, if you have DOS 3.3 then there is an extra bonus that you get with the purchase of a RAMcard, Programmer's Aid #1.

You Apple owners who have had your computers for a long time may recall that in the days when Apples primarily ran Integer Basic, there was an extra ROM chip that you could buy, called the Programmer's Aid #1. This chip plugged into the $D0 ROM position on your motherboard and contained a large number of utilities for Integer Basic, like renumbering, appending from cassette tape, verifying an Integer cassette program, generating musical notes from Integer Basic, and doing high-resolution graphics from Integer Basic. In addition, some non-Integer utilities were included, like verifying a machine language cassette program, a memory test for the Apple, and a program to relocate 6502 machine language programs. This chip was priced at about $50.

Well, the punchline here is that with DOS 3.3 and a RAMcard, you've just gotten that $50 chip for free. It's on your system master diskette for DOS 3.3 as part of the INTBASIC file. When you fire up the HELLO program from the DOS 3.3 system master, the loading of the 50 sector INTBASIC file brings the $D0 ROM chip right along with it. I'm assuming here that most of you probably have Applesoft in ROM on the motherboard. If you don't, you can actually get two copies of INTEGER BASIC into memory by changing the HELLO program slightly so that INTBASIC is BLOADed and then type POKE -16256,0.

When I first became aware of this extra bonus, I realized that without documentation, these features would be impossible to use. So I hurried on down to my local Apple dealer and purchased the Programmer's Aid #1 Installation and Operating Manual, #A2L0011, for about two dollars. I was particularly interested in how the machine language program relocation utility worked. I'm happily moving things all over the place now.

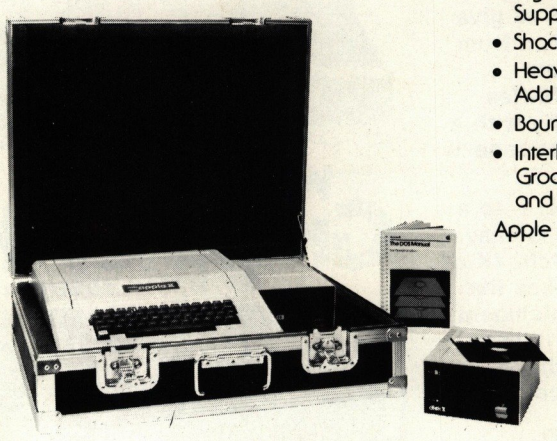
There is another mind boggling possibility which I haven't had time to explore yet. A new assembler called Merlin comes with a wealth of information on Applesoft Basic. By using this info, along with the 50 sector FPBASIC (Applesoft) file on DOS 3.3, and the RAMcard, it is actually possible for you to create you own custom version of Applesoft Basic, with your own special commands and everything. With one of these RAMcards, the possibilities are almost endless.

```
]LIST
5  REM
SO WHAT DID YOU EXPECT?
KEN WATTERS:CALL -A.P.P.L.E.
10  TEXT : HOME
20  FOR I = 1 TO 40:
      PRINT CHR$ (95);: NEXT
30  FOR I = 1 TO 20:
      PRINT  CHR$ (91);:
      FOR J = 1 TO 38:
      PRINT  CHR$ (32);:
      NEXT J: PRINT
       CHR$ (93);: NEXT I
35  POKE 33,34: POKE
      32,3: VTAB 3:
      LIST : POKE 32,0:
      POKE 33,40: VTAB 22
40  FOR I = 1 TO 40:
      PRINT CHR$ (45);: NEXT
50  REM   SO WHAT DID
      YOU EXPECT?
```


FLIGHT FORM

BY CASES, INC.

DESIGNED TO PROTECT YOUR COMPUTER



Features —

- Rigid Shell Made of Plywood Supported High Impact ABS.
- Shock Resistant Foam Lining.
- Heavy Duty Hardware Add $22.50 (U.S. dollars) for lock.
- Bound Metal Edges.
- Interlocking Tongue and Groove Extrusion, Mating Lid and Bottom.

Apple II Plus 2 Drives $175.00

OTHER MODELS AVAILABLE

Overseas Inquiries Welcome

CLUB AND DEALER DISCOUNTS AVAILABLE

CASES, INC. P.O. Box 33820 Seattle, WA 98133-0820 (206) 365-5210

VISA
MASTERCARD

IMAGINE IT...

NEW

Buffer Options Available Up To 64K

CAPTURE IT.

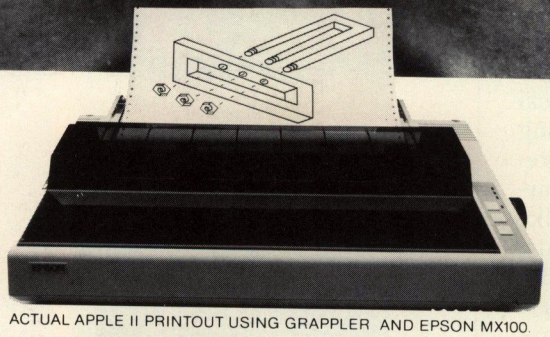
Completely Redesigned. Now, the Grappler+.
The original Grappler was the first graphics interface to give you hi-res screen dumps from your keyboard. The new Grappler+ with *Dual Hi-Res Graphics* adds flexibility with a side-by-side printout of page 1 and page 2 graphics.

Interfacing the Grappler+ to a wide range of printers is easy as changing a dip switch. 4K of exclusive firmware makes the Grappler+ the most intelligent, full-featured Apple® Printer Interface made. And, the Grappler+ is Apple III compatible.*

Up to 64K Buffer Option
An optional Bufferboard can now be added to all existing Grappler and Grappler+ interfaces. See your Apple Dealer for details.

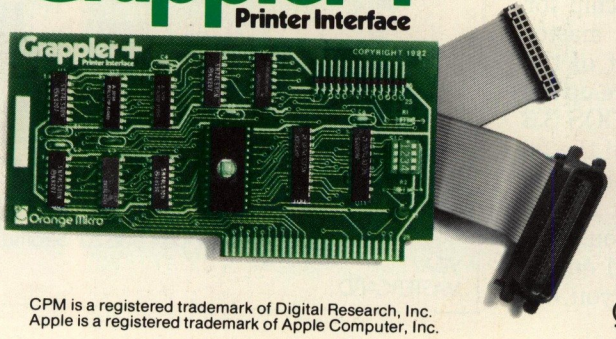
*Requires additional software driver.
**Requires graphics upgrade.

©Orange Micro, Inc. 1982



ACTUAL APPLE II PRINTOUT USING GRAPPLER AND EPSON MX100

With The Grappler™+ Printer Interface



CPM is a registered trademark of Digital Research, Inc.
Apple is a registered trademark of Apple Computer, Inc.

The Grappler+ Features:
• Dual Hi-Res Graphics • Printer Selector Dip Switch • Apple III Compatible* • Graphics Screen Dump • Inverse Graphics • Emphasized Graphics • Double Size Picture • 90° Rotation • Center Graphics • Chart Recorder Mode • Block Graphics • Bell Control • Skip-over-perf • Left and Right Margins • Variable Line Length • Text Screen Dumps.

The Grappler+ also works with Pascal and CPM.

The Grappler+ interfaces with the following printers:
• Anadex • Centronics • Datasouth • Epson** • NEC • C.Itoh • Okidata**

The original Grappler is available for IDS 460, 560, Prism, Microprism.

Orange Micro Inc.
3150 E. La Palma, Suite G
Anaheim, California 92806
U.S.A.
(714) 630-3620
Telex: 183511 CSMA
Foreign Dealer Inquiries Welcome

GET-MOST WRAP-UP:

YES, YOU REALLY CAN PUT FILES ON TRACK 0

Clay Ruth /Call —A.P.P.L.E. Staff Writer

IF you've made the DOS modifications published in "Get the Most from a Diskette" (March 1982) and "Get Most Follow-Up" (April 1982), no doubt by now you have discovered that placing files on track 0 isn't as easy as I said it was!

When I tested my original patches, I moved files to my DOS-less diskette by using FID (from the DOS 3.3 System Master). At that time I didn't realize that FID has its own track/sector allocation routine which works quite differently from the one within DOS. FID readily places files on track 0. With some additional patches, DOS will also treat track 0 on an equal basis with the other 34. These additional patches are summarized in Fig. 1. Here's what is happening:

The biggest obstacle to using track 0 was the "current track allocation" byte at $B5F1 (all addresses in this article refer to a 48K system). Standard DOS 3.3 interprets a zero at $B5F1 as meaning no track is currently allocated. This makes DOS start searching elsewhere for free space. To overcome this, I changed every subroutine that uses this byte, making them interpret $FF (rather than $00) as an indication of no current allocation. This involved changing "BEQ" instructions to "BMI" at $B247 and $B279, and changing a "BNE" to "BPL" at $B2C6. It also required placing $FF at $B266 and at $B2D6. I then intercepted the routines that clear the allocation table ($B5F0-B5F5), making them place $FF at $B5F1. These intercepts occur at $AC02 and $AD91, using patches at $BCF2 and $B6C5 respectively.

The next problem was the baseline DOS uses to index into the VTOC bit maps. The normal baseline is $B3F6, which is just under the track 1 bit map image in RAM. Accessing the track 0 bit map requires a baseline of $B3F2. I changed the $F6 to $F2 at $B2A2, $B2AD, $B2F6 and $B2F9.

The new baseline requires a new calculation of the track offset. I intercepted the calculations at $B29B and $B2EB, using a patch at $BA7C.

The catalog display still wasn't right. My April patch took care of DOS's search for a specific file, but the full catalog display still interpreted track 0 as an end mark. I had to intercept the routine at $ADD7, sending it to $BA83, where a new patch allows rejection only when both the track and sector pointers for the file's T/S list are zero.

Fig. 1 – Modifications required in addition to those published in March and April, 1982, if files are to be placed on Track 0.

Note: *These all go on the Bootable Disk*

a. For Track 2, Sector 1

```
47:30
66:FF
79:30
9B:20 7C BA
A2:F2
AD:F2
C6:10
D6:FF
EB:20 7C BA
F6:F2
F9:F2
```

b. For track 1, Sector E

```
78:4C CE B6
```

c. For Track 1, Sector C

```
65:4C B3 B6
91:4C C5 B6
D7:4C 83 BA EA
```

d. For Track 1, Sector B

```
02:4C F2 BC
```

e. For Track 0, Sector 4

```
7C:A8 C8 98 0A 0A A8 60
83:30 0A D0 05 BD C7 B4
F0 06 4C DB AD 4C 25 AE
4C 2C AE
```

f. For Track 0, Sector 6

```
F2:8D FA B5 CE F1 B5 60
```

g. For Track 0, Sector 0

```
B3:48 F0 03 4C 68 AD C8
B1 42 F0 03 4C 6B AD 68
4C 70 AD
C5:9D F0 B5 CA 10 FA 4C
F5 BC
CE:AA F0 03 4C 7B AF C8
B1 42 F0 03 4C 7E AF 88
4C 82 AF
```

The standard DELETE routine steps its way through the T/S list, resetting bits in the VTOC accordingly. It stops when it sees a track 0 entry. Also, a link to a second T/S list on track 0 is interpreted as no link at all! In both cases, I had to implement the now-familiar process of checking the sector reference too, allowing rejection only when both track and sector are zero. I did this by intercepting at $AD65 and $AF78, using patches at $B6B3 and $B6CE respectively.

By now I have used up just about all the unused space within DOS. If you want to apply other DOS patches that have been published, I've probably already occupied their assigned areas. But there are other options. You might have noticed that three of my patches reside on Page $B6, which is the image of track 0, sector 0. This is usually thought of only as

GAME DESIGNER
You Have the Inspiration
We Have the Tools
introducing
THE FROB™

The hardware/software system that converts your Apple® II into a *sophisticated Atari® 2600 VCS™ game development workstation.*

Now Available for the Atari 5200 HES™

Five minutes to install
No machine modification
Operational simplicity
Real time game development

Order Now For Immediate Delivery.
CALL 408-429-1552

TM

FROBCO, a Div. of Tri-Comp Polytechnical, Inc.
P.O. Box 2780
Santa Cruz, CA 95063

The Miracle of Creation Can Be Yours™

FROB is a trademark of **FROBCO.** Atari and Apple are registered trademarks of Atari, Inc. and Apple Computers, Inc. respectively. VCS and HES are trademarks of Atari, Inc.

PUTTING FILES ON TRACK 0

Boot 1 image. Actually, Boot 1 only uses the first 93 ($5D) bytes and the last two. Bytes $5E through $B2 are Apple's own patches to DOS. Beyond my patches, it is still blank. You can relocate some patches to this area if you like. Just stay clear of bytes $FE and $FF – they will be changed during booting!

Another possible area for patches is the final 48 ($30) bytes of the VTOC. All those zeros from byte $D0 through $FF serve no purpose. Any patch placed here must go on your DOS-less data diskettes as well as on your modified DOS-containing diskettes so it will be available whenever needed. Failure to do so will result in a BRK and subsequent 6502 register dump somewhere between $B48B and $B4BA. These addresses correspond to bytes $D0 through $FF of the VTOC sector; its image in RAM overlaps a page boundary.

I apologize for the incompleteness of my original modifications. I am confident that, with the additional modifications presented here, you will have a DOS that makes trouble-free use of track 0. If you should discover a problem I have missed, please get in touch with me via Val on the hot line.

I wish to thank reader Jack Lewis for first bringing some of these problems to my attention, and for his assistance with developing and testing some of the patches.

```
]LIST
10 A$ = "!/-\":B$ =
     "-/!\"
15  REM
SO WHAT DID YOU EXPECT?

AUTHOR ANOM.
SUBMITTED BY JONATHAN
     KANDELL
CALL -A.P.P.L.E.
20  POKE 2060,92: POKE
     2070,92
30  HOME
40  FOR I = 1 TO 4:
     VTAB 10: PRINT
      MID$ (A$,I,1):
     PRINT  MID$ (B$,I,1)
50  FOR J = 1 TO 33: NEXT
60  NEXT
70  GOTO 40
```

The following lines may be placed at the start of any Applesoft program to POKE all required DOS patches into RAM rather than modifying the DOS on your bootable diskette. They incorporate all patches made in the three articles (Get the Most from a Diskette, Get Most Follow Up, and Get Most Wrap Up).

It is suggested that this be used as the start of the greeting program on your bootable disk. However, line 22 changes all the DATA Statements to REMarks after they are used, so it may also be attached to the beginning of a program that uses other DATA Statements.

```
Command :?PR#0
]LIST

0  DATA  200,177,66,208,7,136,173
     ,187,181,76,8,177,136,177,66
     ,76,20,177,141,250,181,206,2
     41,181,96
1  DATA  189,198,180,240,3,76,243
     ,177,189,199,180,240,3,76,24
     5,177,76,18,178
2  DATA  168,200,152,10,10,168,96
     ,48,10,208,5,189,199,180,240
     ,6,76,219,173,76,37,174,76,4
     4,174
3  DATA  72,240,3,76,104,173,200,
     177,66,240,3,76,107,173,104,
     76,112,173
4  DATA  157,240,181,202,16,250,7
     6,245,188
5  DATA  170,240,3,76,123,175,200
     ,177,66,240,3,76,126,175,136
     ,76,130,175
10 P =  PEEK (987):S = 256 * P +
     1504
11 A = S:B = S + 24: GOSUB 20
12 A = S - 631:B = S - 588: GOSUB
     20
13 A = S - 1581:B = S - 1537: GOSUB
     20
15  POKE S - 3035,76: POKE S - 30
     34,224: POKE S - 3033,P + 5:
     POKE S - 2802,76: POKE S -
     2801,105: POKE S - 2800,P +
     3
16  POKE S - 2638,169: POKE S - 4
     318,76: POKE S - 4317,242: POKE
     S - 4316,P + 5: POKE S - 343
     2,76: POKE S - 3431,206: POKE
     S - 3430,P - 1
17  POKE S - 3963,76: POKE S - 39
     62,179: POKE S - 3961,P - 1:
     POKE S - 3919,76: POKE S -
     3918,197: POKE S - 3917,P -
     1: POKE S - 3849,76: POKE S -
     3848,131: POKE S - 3847,P +
     3: POKE S - 3846,234
18  POKE S - 2713,48: POKE S - 26
     82,255: POKE S - 2663,48: POKE
     S - 2629,32: POKE S - 2628,1
     24: POKE S - 2627,P + 3: POKE
     S - 2622,242: POKE S - 2611,
     242
19  POKE S - 2586,16: POKE S - 25
     70,255: POKE S - 2549,32: POKE
     S - 2548,124: POKE S - 2547,
     P + 3: POKE S - 2538,242: POKE
     S - 2535,242: GOTO 22
20  FOR X = A TO B: READ Y: POKE
     X,Y: NEXT : RETURN
21  POKE LS + 4,178:LS =  PEEK (L
     S) + 256 *  PEEK (LS + 1): RETURN

22 LS =  PEEK (103) + 256 *  PEEK
     (104): FOR X = 0 TO 5: GOSUB
     21: NEXT
```

S&H Software is Speeding up The Apple®!

The DOS Enhancer (TDE) works up to 500% faster than standard Apple DOS 3.3...$69.95

TIRED OF WAITING . . . for your programs to load or save? Then S&H's TDE, licensed by Apple, is the answer.

TDE program updates standard Apple DOS 3.3 disks — or creates copyable TDE disks — with TDE's QuickDOS and Quick-load features.

TDE's QuickDOS runs and saves BASIC and binary programs up to 5 times faster* and is completely compatible with standard Apple DOS 3.3 programs.

TDE "Quick-loads" the RAM card with FPBASIC/INTBASIC, QuickDOS or user program in 1.7 seconds at startup.

TDE "package" includes utility disk, training/support disk, step-by-step instruction manual, S&H's Supercat/menu, and multidrive "Quick-copy" program.

TDE system requirements: 48K Apple][or][+, ROM/RAM card, DOS 3.3 and one or more disk drives.

Here's what the critics say:

- John Mitchener of PEELINGS II: "The speed increase with **TDE** is awesome and is probably worth the price of the program alone, without all the other features . . . AA rating."
- Val Golding, Editor of Call-A.P.P.L.E.: "**(TDE)** stands as a shining example of how utility and application programs take into account every possible system configuration."
- Clark Congleton of The Apple Orchard: "The Quick-load capabilities will make this package attractive to anyone who spends a lot of time at the keyboard."
- Chuck Carpenter of INFOWORLD: "Results in a disk that will boot — very fast — in any Apple system."

*To achieve speeds even faster than a hard-disk drive, combine **TDE** with Axlon's RAMDISK 320K Memory System.

Amper-Sort/Merge (A-S/M II) works up to 1000% faster than even VisiCorp's VisiFile program...$69.95

A-S/M II is the fastest "file clerk" you've ever met. Of all the sort utilities developed to manage Apple][data files none does the job nearly so fast as A-S/M II!

A-S/M II's new features include: S&H's superfast VisiFile index sort (callable from within VisiFile for effortless use), an equally fast S&H random access file index sort, and parameter file editing.

A-S/M II can sort/merge from one to five unsorted files into a single file of up to 125K in size per disk.

A-S/M II's "package" includes: utility disk, training disk, step-by-step instruction manual, and S&H's new Supercat/menu.

A-S/M II's system requirements: 48K Apple][with ROM or RAM card or 48K Apple][+ with DOS 3.3 and Disk II.

Dealer inquiries invited

Apple is a registered trademark of Apple Computer, Inc.

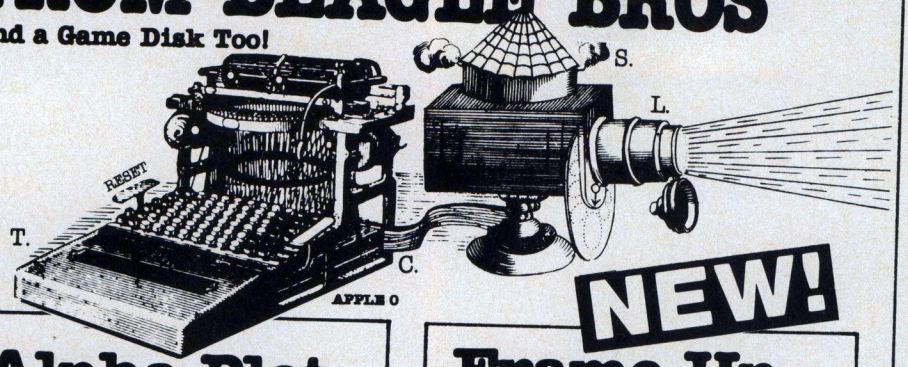
S&H Software

Available from your dealer. **Mail Order:** Send checks to S&H Software, 58 Van Orden Rd., Harrington Park, NJ 07640. 201-768-3144

Credit Cards: Phone Cybertronics International at 212-532-3089
(Overseas Airmail: Add $10.00 postage and handling)

NEW APPLE® UTILITIES FROM BEAGLE BROS

And a Game Disk Too!



NEW!

Alpha Plot

Hi-Res Graphics/Text Utility by Bert Kersey & Jack Cassidy

Here are a few of Alpha Plot's useful graphics features. Compare with other graphic utilities at any price—

HI-RES DRAWING: Create hi-res pictures and charts with text, on both pages; all Optional Xdraw cursor (see lines before drawing). Mix colors & Reverse (background opposite). Circles, Boxes, Ellipses; filled or outlined. **Compress Hi-Res to 1/3 disk space.** Relocate any portion of an image anywhere on either page. Superimpose too & convert hi-res to lo-res for colorful abstracts!

HI-RES TEXT: Beautiful upper/lower case with descenders (no hardware required). Color and reverse characters positionable anywhere (no vtab/htab limitations). Professional-looking **proportional spacing** and adjustable character height and letter spacing. Sideways typing for graphs too!

$39.50

- ☐ Unprotected disk (48K min.)
- ☐ Beagle Bros Apple Tip Book #4
- ☐ Peeks & Pokes Chart

NEW!

Beagle Bag!

12 Games on One Big Disk by Bert Kersey

Twelve great games from our classic Beagle Bros collection—TextTrain, Slippery Digits, Wowzo, Magic Pack, Buzzword... Almost all of our "Game Pack" games have been updated and re-released on one jam-packed unprotected disk! **ALSO INCLUDED** is our "Beagle Menu" greeting program (description under "Typefaces" disk on this page).

Compare Beagle Bag with any 1-game locked-up disk on the market today!

All 12 games are a blast, the price is right, the instructions are crystal clear, AND the disk is copyable! You can even list the programs to see what makes them tick!

$29.50 Unprotected. Paddles NOT required. Beagle Menu works with all normal-DOS disks. Includes Peeks/Pokes Chart.

Frame-Up

Graphics Display Utility by Tom Weishaar

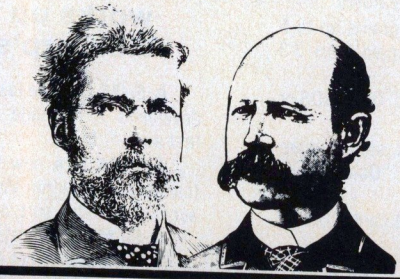
Frame-Up is a very-high-speed Apple "slide projector" utility that lets you create professional-looking displays of intermixed hi-res, lo-res and text pages **on any Apple.** Frame-Up is very easy-to-use and above-all **FAST,** allowing you to load hi-res pictures, for example, in **2½-seconds;** that's five-times faster than normal! Paddles or keyboard are used to change images in forward or reverse order, skipping pages if you want. OR presentations may be left unattended, with **each page individually timed** to appear and remain on the screen from 3 to 99 seconds, as you choose.

Frame-Up includes a sophisticated black and white **text screen editor** that lets you create text "slides" as part of your show. You can even add type "live" on the screen during your presentations. Up to 17 hi-res or 136 lo-res/text pages may be stored per disk. One or two drives are supported. The order and timing of your graphics and text images may be easily (and instantly!) arranged and rearranged. Frame-Up includes a **display module which may be copied** and distributed to your associates so they can run your display, as you designed it, on their Apple or ANY Apple!

Frame-Up is ideal for store displays, presentations to the boss, club programs, trade show booths, product demos, promotions, seminars, conventions, classes, and so on.

$29.50 Machine language. Unprotected. 48K minimum. Peek/Poke Chart included.

BEAGLE BROS DISKS ARE UNLOCKED AND UNPROTECTED. THIS MEANS EVERY PROGRAM IS INSPECTABLE, CUSTOMIZABLE, IF YOU WANT, AND COPYABLE, GIVING YOU THE MOST FOR YOUR SOFTWARE DOLLARS. DON'T SETTLE FOR LESS.



Beagle Bros™ MICRO SOFTWARE

4315 Sierra Vista, San Diego, Ca 92103
714-296-6400

"APPLE" is a registered trade mark of You-Know-Who.

Apple® Mechanic

Shape Writer/Byte-Zap Utility by Bert Kersey

Another best-selling multiple-utility disk—Nine useful, listable, copyable and customizable programs—

SHAPE EDITOR: Put professional hi-res animation in your programs. Keyboard-draw any shape and let your Apple write a shape table and store it on disk. Design large and small **custom typefaces** too, with special characters. 6 fonts on the disk. LIST-able demos show how to use shape tables to animate games, graphic displays, and attractive Charts & Graphs. A valuable time-saving utility/learning tool.

BYTE ZAP: A MUST utility. Rewrite any byte on a disk by loading a sector onto the screen for inspection. **Hex/Dec/Ascii** display optional. Examine bytes via cursor control; enter hex, dec or ascii to change. Create illegal filenames, restore deleted files, change greeting program names, repair/protect disks, change DOS, examine program files. Clear illustrated instructions show how disk data is stored and how to access it. Very educational.

MORE: A disk PACKED with useful music, text and hi-res tricks **for use in your programs.** A great demo-writer program, useful hi-res utilities and educational, entertaining documentation.

$29.50

- ☐ Unprotected disk (48K min.)
- ☐ Beagle Bros Tip Book #5
- ☐ Peeks & Pokes Chart

NEW!

Typefaces for Apple Mechanic

Here are more hi-res fonts for Apple Mechanic's Xtyper and Hi-Writer programs—26 of them at last count, both large and small, all **proportionally-spaced** and positionable anywhere on either hi-res screen. Most are **full 96-character fonts** many with special graphic characters. Each character (from "!" to "□") of every font (from "Ace" to "Zoo-loo") is, of course, editable with Apple Mechanic's Font Editor.

BONUS: Here's BEAGLE-MENU! A unique greeting program that displays **only the catalog file names you want** on the screen (for example, only locked-Applesoft files, or only Binary files) for one-key cursor selection. Just hit Return to Run, Brun or Exec the program at the cursor. Many other features— Space-on-Disk, Load/Bload option, forward and backward catalog "scrolling" for easy file location, and optional sector-number elimination. PLUS the ability to **swap file names** in your catalog!

$20.00 Unprotected. Beagle Bros' Apple Mechanic disk is required to utilize the type fonts. Beagle-Menu works with all normal-DOS 3.3 disks.

If you don't find our products at your Apple Dealer, tell him to phone Beagle Bros, 714-296-6400, **OR** his favorite software distributor.

If you don't find our products at your Apple Dealer, tell him to phone Beagle Bros, 714-296-6400, **OR** his favorite software distributor.

NEW!

Flex Text

70-Column Text Utility
by Mark Simonsen

Flex Text is a unique utility that lets you print variable-width text on Apple's hi-res screens in normal 40-column format, 20-column expanded or 56- and **70-column condensed** characters. Character widths may be mixed as you like for emphasis. Flex Text understands **normal Applesoft** Basic commands, including Home, Inverse, Normal, Vtab 1-24 and **Htab 1 through 70**! It also supports text window pokes and scrolling, so you can program normally, but with the **ability to add text to graphics**, or graphics to text! You can even run your existing programs using these features!

Flex Text is easy to use; just boot it and go! You can now display **upper and lower case** characters in any width without hardware. Every keyboard character may be redefined as any symbol you like with a custom text **character editor.** You may toggle between the "normal" text screen and both hi-res pages if you like. Flex Text is **completely compatible** with Neil Konzen's Program Line Editor and G.P.L.E.

$29.50 Machine language. Unprotected. 48K min. Peek/Poke chart included. Condensed character display requires a monitor (instead of a tv) for best results.

```
10 REM HI-RES NUMBER GENERATOR
20 SIZE=5: SCALE=SIZE: REM NUMBER-HEIGHT
30 HGR: HOME: POKE 232, 0: POKE 233, 3: ROT=0
40 FOR A=768 TO 830: READ B: POKE A, B: NEXT A
50 N=N+1: N$=STR$(N): X=99: Y=0
60 FOR A=1 TO LEN(N$): HCOLOR=0: DRAW 8 AT
   X, Y: HCOLOR=3: DRAW VAL(MID$(N$, A, 1))
   AT X, Y: X=X+SIZE+SIZE: NEXT A: GOTO 50
70 DATA 20, 0, 24, 0, 27, 0, 31, 0, 35, 0, 39, 0, 44, 0,
   49, 0, 52, 0, 57, 0, 53, 62, 36, 0, 49, 38, 0, 53,
   55, 61, 0, 53, 23, 37, 0
80 DATA 46, 38, 52, 0, 61, 46, 62, 5, 0, 61, 54, 37, 7,
   0, 53, 38, 0, 54, 37, 60, 46, 0, 53, 39, 53, 62, 5, 0
```

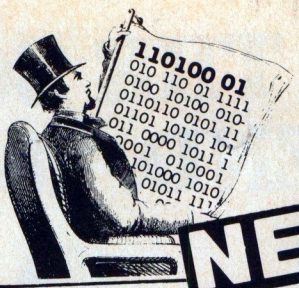
DOS Boss

Disk Command Editor
by Bert Kersey & Jack Cassidy

A classic Apple utility you will ENJOY! Rename DOS commands ("Catalog" can be "Cat", etc.). PROTECT PROGRAMS; any unauthorized save- attempt produces a "Not Copyable" message. Also **List-prevention** and 1-key program-run from catalog. Custom catalogs: Change Disk Volume message to your title; Omit or alter file codes. Rewrite error messages: "Syntax Error" can be renamed "Oops!!" or anything you want! **Two books included**— Fascinating documentation and hours of good Apple reading!

Dos Boss's change features may be appended to your programs so that anyone using your disks (booted or not) formats DOS as YOU designed it.

$24.00
- ☐ Unprotected disk (32K/48K)
- ☐ The Dos Boss Book
- ☐ Beagle Bros Apple Tip Book #2
- ☐ Peeks & Pokes Chart



NEW!

Before you buy any Apple game disk, compare features:

Is there more than one game? Beagle Bag features TWELVE unique games on one disk PLUS a bonus greeting-program utility for use on all of your existing 3.3 disks.

Can it teach programming skills? Beagle Bag games are listable so you can see what makes them work. You can even CHANGE each game's features if you want.

Is the disk copyable? Beagle Bag can be backed-up with ANY copy program. Don't buy software that can't be backed up!

Is the disk unlocked? The Beagle Bag disk can be catalogged, loaded-from, saved-to and Fidded, making it more flexible and more FUN than any locked-up One-Game disk on the market today.

Is the disk priced right? Counting the bonuses, Beagle Bag costs less than $2 per program. This is a disk that will "earn its keep" as long as you own your Apple!

BEAGLE BAG $29.50
(See description on previous page.)

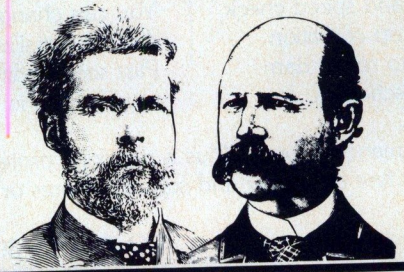
Tip Disk#1

100 Tip Book Tips on Disk
by Bert Kersey

100 programs from Beagle Bros' Tip Books 1, 2, 3 and 4— Fascinating tricks to make your Apple do things it's never done before! All 100 programs are listable, copyable and changeable; and each teaches another fascinating Apple programming technique. **Two different charts are included.**

$20.00
- ☐ Unprotected (32K/48K)
- ☐ Peeks & Pokes Chart
- ☐ Apple II Command Chart

BEAGLE BROS DISKS ARE UNLOCKED AND UNPROTECTED. THIS MEANS EVERY PROGRAM IS INSPECTABLE, CUSTOMIZABLE, IF YOU WANT, AND COPYABLE, GIVING YOU THE MOST FOR YOUR SOFTWARE DOLLARS. DON'T SETTLE FOR LESS.



Beagle Bros™ MICRO SOFTWARE

4315 Sierra Vista, San Diego, Ca 92103
714-296-6400

"APPLE" is a registered trade mark of You-Know-Who.

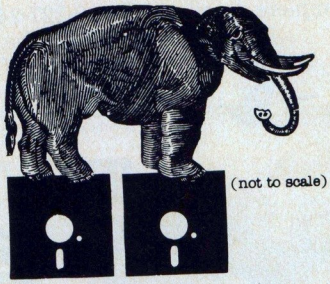
Utility City

21 Utilities on One Disk
by Bert Kersey

A best-seller since it hit the market, and a MUST for your program-development library. Take a look at the features— **List Formatter** makes properly-spaced & indented listings with page breaks; each statement on new line, if-thens and loops called out; a great de-bugger! **Multi-Column Catalog** in any page-width to any printer or CRT. Auto-post Run-Number and last-used Date in programs. Put INVISIBLE working commands in your listings. Access program lines in memory for repair & illegal alteration. Alphabetize & store info on disk. Run any program while another stays intact. Renumber to 65535. Save inverse, trick and **invisible file names**. Convert dec to hex & binary, or Integer to FP. Append programs. Dump text screen to printer...

21 LISTABLE UTILITIES TOTAL!

$29.50
- ☐ Unprotected disk (48K min.)
- ☐ Beagle Bros Apple Tip Book #3
- ☐ Peeks & Pokes Chart


(not to scale)

GOTO Your Apple Dealer.

Most dealers carry our software. If yours doesn't, he can have our disks in his store for you **within 2-3 days** by calling us or his favorite software distributor.

Or Order by Phone:

24-Hour **Toll-Free** Order Desk:
Visa/MasterCard/COD* orders, call
Nationwide: **1-800-854-2003** ext.827
California: **1-800-522-1500** ext.827
Alaska/Hawaii: **1-800-854-2622** ext.827
(ORDERS ONLY, PLEASE) *COD, add $3.00.

OR ORDER BY MAIL—

RUSH! The disk packages checked below plus the free 11x17 Peeks & Pokes Chart:

☐ Alpha Plot	$39.50	☐ Frame-Up	$29.50
☐ A.Mechanic	$29.50	☐ Tip Disk#1	$20.00
☐ Beagle Bag	$29.50	☐ Typefaces	$20.00
☐ DOS Boss	$24.00	☐ Utility City	$29.50
☐ Flex Text	$29.50		

(Add $1.50 Shipping, any size order. California, add 6% tax.)

NAME$= _____

ADDR$= _____

CITY$= _____

ZIP$= _____

VISA — **VISA/MASTERCARD Include Card Number & Expiration Date.** — master charge THE INTERBANK CARD

Mail U.S.check, money order or Visa/MasterCard numbers to:

BEAGLE BROS, Dept. Z
4315 SIERRA VISTA
SAN DIEGO, CA 92103

All Orders Shipped Immediately.

Please add $4.00 for shipping outside North America. COD orders add $3.00. California residents, add 6%.

Apple™ users. THE SOURCE℠ and TRANSEND™. Together for the first time for only $89.

Buy our $89 Transend state-of-the-art data communications software and membership in THE SOURCE, AMERICA'S INFORMATION UTILITY℠, is included. This combination allows you to easily access one of the world's largest information services for up-to-the-minute news and sports, stock prices, travel services and much more.

An optional 260-page Source User's Manual is available for $19.95. Your dealer has complete details. Call 800-227-2400, ext. 912 (in Calif. 800-772-2666, ext. 912) for the name of your local Transend dealer. Offer expires Oct. 1, 1982.

Apple is a trademark of Apple Computer Inc.
THE SOURCE and AMERICA'S INFORMATION UTILITY are service marks of Source Telecomputing Corp., a subsidiary of The Reader's Digest Association Inc.

The Transformation People.

SSM Microcomputer Products Inc.
2190 Paragon Drive, San Jose, CA 95131

Version 4.0

The Ultimate APPLE® Utility Program
COPY II PLUS

Now you can back up your protected software. Copy II Plus is the most sophisticated bit copy program available. It handles synchronized tracks, half tracks, nibble counting, bit insertion and other protection schemes. It also includes a comprehensive discussion of disk format and protection techniques, and instructions on how to back-up dozens of popular programs. A disk system and speed check assure your drives are running in top condition and a nibble editor will allow you to repair damaged diskettes, analyze protection schemes, etc.

The last DOS utility disk you will need. Fully menu driven, the Copy II Plus utilities include a catalog display with binary file addresses and lengths, a disk usage map, and the ability to verify and compare files for differences. It can copy, lock, unlock and delete files and DOS can be copied, or removed from a disk to free up space. You can change the greeting program on a DOS disk, or initialize a disk from scratch. The Copy II Plus sector editor will allow you to view and modify data in either hex or ASCII format.

For your convenience, Copy II Plus is not copy protected for backup.
Available at fine computer and software stores or direct from:

CENTRAL POINT Software, Inc.
P.O. Box 19730-203
Portland, OR 97219
(503) 244-5782

$39.95

Attention current Copy II Plus owners: Return your original disk with $19.95 for an update to Version 4.0.

A CHRISTMAS PRESENT FROM A HACKER

Philip Chien

SO you want to give your friends, customers, associate, enemies, etc. a Christmas, Hanukkah, Winter Solstice, etc. present that says "I'm a hacker, I own an Apple, I forgot to get you a present till the last second, etc."?

How about a calendar with a Hi-Res picture? If you have: a printer with graphics capabilities, an Apple, the Computer Station "Enhanced Graphics Software" or other similar graphics dump programs, and plenty of paper, then you can give a special present.

The following program will dump a Hi-Res picture and print a calendar for any year after 1752 (before September 1752, England and the Colonies were on a different calendar system. Eleven days disappeared; putting any year before 1752 off by eleven or more days). Type in the program and save it onto a DOS 3.3 diskette. Use the FID program (on your 3.3 master disk) to transfer the graphics driver for your printer (i.e. TIGER460.OBJ, if you have an IDS printer model 460, 560, Prism 80, Prism 132, or MicroPrism 480) onto the same disk as the program. While you are in FID, transfer a couple of picture files onto the disk. If you don't have any pictures around that you like, contact your dealer.

Each authorized Apple dealer has two contributed diskettes from Apple (Slide Show 1 and Slide Show 2) filled with Hi-Res pictures. Any decent dealer will, and can, make you a copy of the diskette for a small charge.

Exit FID and type "BRUN DRIVER.OBJ" where DRIVER.OBJ is the name of your driver. Answer the questions concerning your printer setup. Decide what size and color-mode (plot/picture) that you want and dump a picture. Exit the graphics driver program and type "BSAVE DRIVER.OBJ,A$1800,L$800" this will tell the driver the correct size, color, and system configuration when the Calendar program runs it.

If you are using either: a printer interface card with built-in graphics capabilities (i.e. PKASO, Grappler, Practical Peripherals Microbuffer, etc.) or a different graphics driver (i.e. the driver on the Visiplot diskette) then change lines 100 and 270 as necessary. You cannot use the Zoom Grafix Hi-Res dump program because, due to the $!& copy protection system, you can't attach Zoom Grafix to your own program.

If your Apple hasn't been modified for lowercase; then just put in the names of the days of the week and month as capital letters.

If you aren't using a parallel card or Super Serial card, change line 290 to whatever [Ctrl] characters or POKE is necessary for your interface card to set the printwidth.

POKE 36,X is used instead of TAB (X) because a) it's faster, and b) it works (beyond 40 columns, tab won't work properly).

Program Statistics:

Variables

- D$ everyones favorite variable [Ctrl-D]
- S$ Name of Months
- L Length of each month in days
- Z$ Name of Days of Week
- M Month Counter
- T Day Counter
- A$ Name of Picture
- C$ Number of Columns on printer
- C Number of Columns on printer
- D Day of Week counter
- DA Date of Month counter
- A Year requested

Lines

90-150 Setup variables
160-260 Input information and load picture
270 dump Hi-Res picture
280-300 setup top of page
310-330 calculate starting day of year, and leap year if necessary
340-350 print year
360-390 set up each month
400-470 print dates of months
480-510 end year, spit out paper and go again

Source for "Enhanced Graphics Software"

Your local Apple dealer or:
Computer Station
11610 Page Service Drive
St.Louis, MO 63141
(314) 432-7019
Price: approx. $55.

The Program (finally)

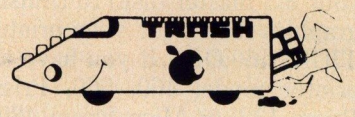
```
]LIST

10  REM

  HI-RES  CALENDAR PRINTER
      BY  PHILIP CHIEN

CALL -A.P.P.L.E. : DEC. 1982

90 D$ =  CHR$ (4): REM  CTRL-D (W
     HAT ELSE?)
100  PRINT D$;"BLOAD DRIVER.OBJ":
      REM  CHANGE GRAPHICS.OBJ TO
      NAME OF PRINTDRIVER
110  DIM S$(12),L(12),Z$(7)
120  FOR M = 1 TO 12: READ S$(M),
     L(M): NEXT
130  DATA  JANUARY,31,FEBRUARY,28
     ,MARCH,31,APRIL,30,MAY,31,JU
     NE,30,JULY,31,AUGUST,31,SEPT
     EMBER,30,OCTOBER,31,NOVEMBER
     ,30,DECEMBER,31
140  FOR T = 1 TO 7: READ Z$(T): NEXT

150  DATA  SUN,MON,TUE,WED,THU,FR
     I,SAT
160  TEXT : CALL  - 936: PRINT : PRINT
     : PRINT "CALENDAR PROGRAM": PRINT
     "( C) 1982 BY PHILIP CHIEN"
170  INPUT "WHAT PICTURE(CATALOG
     FOR CATALOG ,(CR) FOR PICTUR
     E IN MEMORY)?";A$
180  IF A$ = "" THEN 230: REM  PI
     CTURE IN MEMORY
```

A CHRISTMAS PRESENT FROM A HACKER

```
190   IF A$ <  > "CATALOG" THEN 22
      0
200   PRINT D$;"CATALOG"
210   GOTO 160
220   PRINT D$;"BLOAD";A$;",A$2000
      "
230   GR : POKE  - 16297,0: REM  S
      ET GRAPHICS MODE, AND SHOW P
      ICTURE
240 C = 40: REM  DEFAULT PRINTER
      WIDTH
250   INPUT "YEAR?";A
260   PRINT "COLS ON PRINTER 40"; CHR$
      (8); CHR$ (8); CHR$ (8);: INPUT
      C$:C =  VAL (C$): IF C = 0 THEN
      C = 40
270   CALL 6147: REM  DUMP PICTURE

280   PRINT D$;"PR#1": REM  THE PR
      INTER BETTER BE IN SLOT 1!
290   PRINT  CHR$ (9);C + 1;"N": REM
       THIS SETS UP A PARALLEL CARD
       FOR C+1 COLUMNS OF TEXT
300   PRINT : PRINT : PRINT
310 D =  INT ((365.25 * A / 7 -  INT
      (365.25 * A / 7)) * 7 - 1.25
      )
320   IF D < 1 THEN D = D + 7
330   IF A / 4 =  INT (A / 4) AND
      A / 100 <  > INT (A / 100) OR
      A / 400 =  INT (A / 400) THEN
      L(2) = 29: REM  LEAP YEAR?
340   POKE 36,(C / 2 - 7): PRINT "
      ****";A;"****": REM  PRINT Y
      EAR
350   PRINT
360   FOR M = 1 TO 12
370   POKE 36,(C / 2 -  LEN (S$(M)
      ) / 2): PRINT S$(M): REM  PR
      INT MONTH
380   PRINT
390   FOR T = 1 TO 7: POKE 36,(T *
      C / 7 - 3): PRINT Z$(T);: NEXT
      : REM  PRINT DAYS OF WEEK
400   PRINT : PRINT
410   FOR DA = 1 TO L(M)
420 D = D + 1: POKE 36,(D * C / 7
      - 3)
430   IF DA < 10 THEN  PRINT "0";
440   PRINT DA;
450   IF D = 7 THEN D = 0: PRINT :
       REM  NEW LINE
460   NEXT DA
470   PRINT : PRINT
480   NEXT M
490   PRINT  CHR$ (12): REM  NEW P
      AGE
500   PRINT D$;"PR#0"
510   GOTO 160
32767  REM  CALENDAR  PROGRAM BY
     THE APPLE WIZARD
32768  REM  COPYRIGHT 12/26/79
32769  REM  LAST MOD 9-27-82
```


REPLAY

APPLE PROGRAM COPY SYSTEM



DISK FORMATING IRRELEVANT

- **COPIES TOTAL LOAD PROGRAMS**
- **60 PAGE MANUAL INCLUDED**
- **COPY IN 20 SECONDS**
- **RESTART IN JUST 10 SECONDS**
- **CAN ANALYZE PROGRAMS AND DISKS**
- **WILL PACK AND CONDENSE COPIED PROGRAMS TO BRUNNABLE DOS FILES**
- **NO LANGUAGE CARD NEEDED FOR PACKED FILES**

TO ORDER:
Write or call
Texas Ranch And Shoreline Systems
5712 Manor Road, Box 20
Austin, TX 78723
(512) 926-4527

COST:
$129.95 Fully Assembled Card
Plus $5.00 Shipping and Handling
Outside U.S. add $5.00 Air Parcel Post
(Texas residents add 5% sales tax)
VISA/MASTERCARD WELCOME!

REPLAY is an interface card that plugs into any Apple* slot. REPLAY does not copy a disk, rather it copies a program executing in memory. The original disk formating is not important.

NOW game players can save a game at ANY LEVEL and QUICKLY restart it with the REPLAY card.

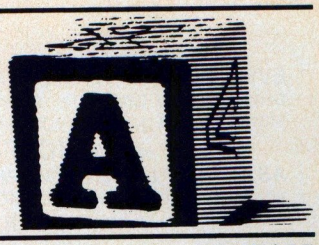
The copied program does not need the REPLAY card to execute. Two copied programs can be put on one DOS 3.3 disk. That disk is ALL that is needed to run the program.

With the packing program supplied, the copied program can be packed and Brunned from DOS 3.3 without a language card. Utilities and documentation are supplied with the system. Tutorials are given on multi-access disk analysis, copying and packing.

*Apple is a registered trademark for Apple Computer, Inc.

A IS FOR APPLE:

PRINT ASCII



Ralph H. Swerdlow, M.D./Call —A.P.P.L.E. Staff Writer

WOULD you like to have a little program that could tell you the ASCII of any symbol that you can type at the keyboard, and also tell you what CHR$(N) is for any N?

Here is such a program. It can save you the chore of looking for this information while you are programming. If you save it on your "UTILITY DISK" it will be available for your use any time you need it.

If you have only one disk drive, you can save this program on the disk that you are using while you develop your program by inserting your utility disk, and typing, "LOAD CHRASCII". Then remove the utility disk, insert your working disk, and type "SAVE CHRASCII." Then, whenever you need to know, save your working program, run "CHRASCII," and you will have the information. With two disk drives, save the program you are developing on one disk, and run "CHRASCII" from the other drive.

```
]LIST

10  TEXT : HOME
20  VTAB 5: PRINT "WHICH
      DO YOU WANT?:"
30  VTAB 10: HTAB 10:
      PRINT "(1) ASCII"
40  VTAB 12: HTAB 10:
      PRINT "(2) CHR$(N)"
50  VTAB 16: HTAB 15:
      PRINT "YOUR CHOICE";:
      INPUT A
60  IF A = 1 GOTO 100
70  IF A = 2 GOTO 200
80  IF A <  > 1 AND
      A <  > 2 THEN
       CALL  - 198:
      GOTO 10
100  HOME : VTAB 4:
      INPUT "TO FIND
      ASCII, ENTER LETTER:";
      L$
110  VTAB 8: HTAB 5:
      PRINT "ASC (";L$;")=
      "; ASC (L$)
120  VTAB 12: HTAB 10:
      PRINT "HIT ANY
      KEY FOR MORE.....":
      GET C$: GOTO 100
200  HOME
210  VTAB 5: HTAB 5:
      INPUT "ENTER N
      :";N
220  VTAB 10: HTAB 5:
      PRINT "CHR$(";N;")
      = "; CHR$ (N)
230  VTAB 15: HTAB 10:
      PRINT "FOR MORE,
      HIT ANY KEY....":
      GET C$: GOTO 200
```

Line 10 clears and sets the screen to full width. Lines 20 to 50 ask you which you would like. The VTABS and HTABS in these lines format the screen.

In line 50, you are asked for your choice. Note that a semi-colon occurs after the last quotation mark. This prevents a line feed, and the input request with its question mark is printed on the same line.

Lines 60 and 70 direct the program to line 100 or 200 depending on which key is hit, the "1" or "2" key.

Line 80 contains the command "CALL -198," which will ring the Apple's bell, and says, "If any key other than the 1 or 2 keys are pressed, sound the bell and start over again."

Line 100 begins the part of the program in which the user enters any character that can be typed. A string, L$ is formed. Study line 110 carefully because it may be confusing at a first glance. It prints what is between the first two quotation marks ASC (then L$, then the characters between the second set of quotation marks, and finally, the ASC of L$. The semi-colon concatenates, or joins these together, by preventing the line feed. Line 120 gives the user a chance to find the ASC of additional characters, if so desired, by going back to line 100.

Lines 200 to 230 clear the screen, ask for a number, and then print CHR$(N).

When you are through with this program, hit the RESET key, type, "NEW," and then continue with whatever you wish to do.

You can also use any of the nice "RENUMBER" programs and place this program high in memory, out of the way of the program you are working on, and then use it any time you wish to.

The range of N is 0 to 255, but remember that if N is less than 32, CHR$(N) is a control character and will not print. If N is greater than 192, and you have a lower case chip, you will get lower case letters printed.

Would you like a list of CHR$(N) printed for you to study?

Here is a neat program that will do it for you.

```
]LIST

10  TEXT : HOME
20  C = 0
30  FOR N = 1 TO 255
40  C = C + 1
50  PRINT "#";N;"=";
      CHR$ (N),
60  IF C = 60 THEN C
      = 0: PRINT "ANY
      KEY": GET C$:
      HOME
70  NEXT N
```

Line 10 should be obvious. Line 20 sets C to 0. This will be used to print CHR$(N) one screen full at a time. Line 30, starts a "FOR-THEN" loop, increasing N from 1 to 255. Line 50 prints, "#N = CHR$(N)." Note the use of the comma which ends line 50. In Applesoft, the comma acts somewhat like a tab, printing out into columns. Line 40 causes C to be increased each time the program loops. When C = 60, twenty lines have been printed to the screen, and line 60 then stops the program, sets C back to 0, and then asks you to press any key. The screen is then cleared and the next group of CHR$(N) will be printed in a tabbed format three columns across your screen.

If you have a printer, you can use this program to print these out so that you have a permanent hard copy to review at your leisure by adding line 5:

```
5 D$ = CHR$(13) + CHR$(4):
  PRINT D$;"PR#S"
```

with S equal to the Slot number in which you have your printer.

By studying these two simple programs and pages 8 and 15 of *The Apple][Reference Manual,* and page 138 of the *The BASIC Programming Reference Manual,* your knowledge of the ASCII representation of characters, as well as your computing skills, will be greatly increased.

NEW FROM NORTHERN!

EVER WONDER WHY YOUR APPLE GIVES YOU 9^3=729.000001 INSTEAD OF 729? The answer to this and many other strange properties of computer arithmetic is contained in ***COMPUTER ARITHMETIC FOR THE APPLE USER.*** If you really want to understand what is going in your computer you can't afford to overlook this book.
$15.95

AND IF YOU'RE JUST GETTING STARTED IN ASSEMBLY LANGUAGE . . .

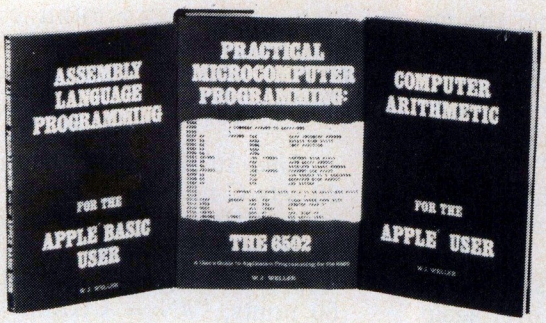
ASSEMBLY LANGUAGE PROGRAMMING FOR THE APPLE BASIC USER starts at the beginning, relating assembly language to your experience with BASIC. It contains a number of useful example programs that you can modify including a simple music generator. If you're looking for a place to get started this is it.
$15.95

AND THE LAST WORD . . .

The ultimate authority on assembly language programming for the Apple or any other 6502 based computer is ***PRACTICAL MICROCOMPUTER PROGRAMMING: THE 6502.*** In 20 chapters and 480 pages this book covers all of the assembly language technique you will need to create your own *fast* machine language programs, free of the limitations of BASIC. It includes the full source text of an Apple/6502 assembler and debugging monitor that runs on any Apple II or Apple II+. You get the object code FREE, that's right, FREE, when you return the coupon at the back of the book to us. You couldn't buy the software alone at twice this price.
$32.95

AND A FREEBIE!

If you order any two of the above books with this coupon you will get our special monograph ***ULTRA HIGH SPEED GRAPHIC TECHNIQUES FOR THE APPLE II*** at no charge. This publication shows how to do *really* fast graphics with assembly language programs. It is not for sale. It is available only with the purchase of any two of the three books above.



Mail to: **Northern Technology Books**
Box 62
Evanston, IL 60204

- ☐ Computer Arithmetic for the Apple User $15.95
- ☐ Assembly Language Programming for the Apple Basic User $15.95
- ☐ Practical Microcomputer Programming: the 6502 $32.95
- ☐ **Check enclosed (U.S. funds only)**
- ☐ **Money order enclosed**

Name _____

Street _____

City _____ State _____ Zip _____

Illinois residents add 6% sales tax.

```
]LIST
10  REM

SO WHAT DID YOU EXPECT DEPT

JONATHAN KANDELL-CALL -A.P.P.L.E.

13  HOME : LIST
100  CLEAR
110 A =  RND (1) * 22 + 1:B =  RND (1) * 22 + 1
120  VTAB A:C =  PEEK (40) +  PEEK (41) * 256:
     VTAB B:D =  PEEK (40) +  PEEK (41) * 256
130  FOR I = 1 TO 39:A$ = A$ +  CHR$
     ( PEEK (C + I)):B$ = B$ +  CHR$
     ( PEEK (D + I)): NEXT
140  VTAB A: PRINT B$: VTAB B: PRINT A$: GOTO 100
```

A.P.P.L.E. Members only

Be the first on your block!

with

Kids & The Apple

Kids & The Apple is an 8½ x 11 full color tutorial textbook written with **clear and easy instructions** at the 10-14 year old age level.

Kids & The Apple will teach kids to program board games, word games, action games, how to store & recall useful data, write drill programs for math, spelling, history and illustrate them with Apple's graphic capabilities. All this in 33 carefully sequenced lessons with explanations, examples, exercises and review questions.

$16.00

A.P.P.L.E. ORDERS
304 Main Ave. South
Renton, WA 98055
(206) 271-4514

Washington State Residents add 7.0% sales tax
Foreign shipping/handling charge: $6.00
Canadian postage being revised

DOUBLE your DISKETTES

The only reasons your APPLE II cannot use the back side of your diskette are:

1. There is no notch.
2. The diskette manufacturer either did not test the back side, or worse, put the flawed side to the back.

A pair of diagonal cutters will solve problem number 1.

DISK PREP

will solve problem number 2.

DISK PREP formats and tests your disk. Sectors with flaws are left so they cannot be used. Your disk is left ready to boot, complete with a flaw report program saved on it.

$25.00

sympathetic software
9531 Telhan Drive
Huntington Beach, Ca. 92646
DOS 3.3 and DOS 3.2.1
California residents add $1.50 tax.

A Call —A.P.P.L.E. Review:

PASCAL FROM BASIC

David G. Sparks/Reviewer

Addison-Wesley Publishing Company
$12.95 (1982 182 pages)

MOST of the readers of this magazine are beyond the beginning stages of their experience with the Apple. A good many readers are quite proficient, and turn to these pages to refine skills and knowledge already hard-won through experience.

Thus it is a pleasure to introduce this new book about Pascal. For the first time in this reviewer's experience, someone has produced a guide to Pascal which assumes the reader already is an able programmer in BASIC.

Brown even begins by pointing out the things we always believed were BASIC's strong points: superior string handling, straightforward program entry, line-numbered layout, ease of program editing and testing, direct memory manipulation via PEEK and POKE, no need to deal with an awkward-seeming operating system (just turn'er on and go), and last but not least, our very familiarity with BASIC, which makes us wonder why we should ever need to learn any other language.

He honestly admits that the appearance of a Pascal program to someone raised on BASIC is unnerving, if not downright bizarre. He includes a delightful cartoon which says, "Using (a well-known operating system) is like kicking a dead whale along a beach."

But the book is about Pascal, after all, so Brown valiantly presses on to discuss the merits of the language. What keeps the reader going, at least at first, is the delightful style of writing which commences at the beginning and carries pleasantly along to the very end.

It is already clear that this book does not tout Pascal as the be-all and end-all of computer science. But Brown believes that Pascal is worthy of any serious programmer. This candid, even-handed approach makes his the first Pascal book which this BASICally biased reader truly enjoyed, and read at one sitting, for that matter.

He uses the reader's ability with BASIC as a teaching aid. Programming concepts already familiar in his reader's "native" language are translated into fully idiomatic, fluent Pascal. As Brown puts it, his purpose is "Changing the way you think... about solving programming problems. The mechanisms offered by Pascal...allow you to set about problems in radically new ways. Thus when we talk about "learning Pascal" we mean a lot more than learning the grammatical and semantic rules of the language."

In an effort to be "serious without being solemn," Brown's book is populated with charming characters. Two of them come along quite early.

"Bill Mudd has been programming in BASIC for twenty years. One of his programs is 10,000 lines of BASIC, and it works — at least most of the time. His comments... were not encouraging.

"Pascal looks to me like a language for head-in-the-clouds academics, who have never written a real program in their lives," he said. [And Bill went on to say a great deal more which was, to say the least, unflattering.]

"Rather upset by Bill's uncouth attitude, we consulted our other friend, Professor Primple. The Professor is fanatically keen on Pascal. He has, like Bill, had twenty years of writing programs. He has written solutions to the eight queens problem in thirty different programming languages,... and, best of all, has encoded Ackermann's function in seventy-two languages.

"Converting the heathen is always worth doing, I suppose,' he told us, scratching his little beard in his characteristic way..."

Bill and the Professor pop up repeatedly. They are joined by Mrs. Buzz, Perkins the security guard, along with the two bad guys of programming, Mr. 869704 (who thinks the whole world can be described by a sufficient quantity of numbers) and Frank Round (whose name came out of an automatic language translating program, when trying to interpret "open circuit"). Honest, this is a serious book!

Brown argues that Pascal's peculiarities are among its great strengths. That it requires variables to be declared and typed separately from the body of the program is good, he argues, because this makes sure that variables aren't scattered like breadcrumbs throughout the program text. Bugs, the Professor might suggest, are better squashed when bunched together.

The structuring of data is much more elaborate in Pascal than in BASIC, causing many humorous hoots of derision from Bill Mudd. In addition to integers, reals, characters, and strings, Pascal employs arrays and something called "records" to build new data types capable of considerable sophistication. Brown gives the example of complex numbers, in which each value has both a real number component, and an imaginary part.

Pascal allows a TYPE declaration such as

```
TYPE COMPLEX = RECORD
        REALPART : REAL;
        IMAGINARY : REAL;
        END;
```

The program would also contain a VAR declaration in which A, B, and C are defined to be of type COMPLEX. Each component of A is addressed separately, as in

```
A.REALPART := 17.5;
A.IMAGINARY := 3.12;
```

The programmer might write a procedure to add two such complex variables and report the sum in a third. Within the program, the call to the procedure, with an enlightening comment, might appear as

```
ADDCOMPLEX (A, B, C);
(* C = A + B *)
```

This writer's fantasy is a TYPE declaration in which a single variable type is built up from many sub-types, which holds all the information one could possibly imagine about a book. Follow along, if you please...

First we must build from rudimentary data types. Therefore

```
TYPE
   statement    = string;
   pagenum      = 1..999;
   chapnum      = 1..25;
```

—●→

leading to a slightly more complex

```
sentence      = array[1..50] of
                statement;

  (*Sentence actually describes an
  entire page, since it is an array of
  up to 50 single statements. Any
  given statement can be extracted
  by specifying "sentence[X]".*)
```

As we build, we get into less familiar territory, but it is still relatively easy to follow what is going on.

```
humanname =
  record
    first : string;
    middle : string;
    last : string;
  end;
(* We can thus sort by either first
or last name. *)

chapter       =
  record
    chapname : string;
    firstpage : pagenum;
    lastpage : pagenum;
  end;
```

and finally

```
BOOK          =
  record
    title : string;
    author : humanname;
    numofpages : pagenum;
    numofchapters : chapnum;
  contents:
    array [1..25] of chapter;
  page :
    array [1..999] of sentence;
  end;
```

Contemplate a variable named WORTHYBOOK of TYPE BOOK. One could assign WORTHYBOOK.TITLE := "Pascal from BASIC." WORTHYBOOK.NUMOFPAGES := 182. WORTHYBOOK.AUTHOR.LAST := "Brown." Properly filled in with its data, WORTHYBOOK is a wealth of information collected into one variable. A command to WRITE (WORTHYBOOK.CONTENTS [8].CHAPNAME) would print the title of chapter 8, "Records." A futuristic, talking computer might even be able to comply with the instructions in a user-programmed procedure to READALOUD (WORTHYBOOK, WORTHYBOOK.CONTENTS [8].FIRSTPAGE, WORTHYBOOK.CONTENTS [8].LASTPAGE)!

Our imaginary variable, WORTHYBOOK, contains so much information that it would not fit on a single diskette. Yet, if the storage peripheral were capacious enough, the entire mass of information, both string and numeric, could be written to a file called SHELF with a single PUT(SHELF) command, and later retrieved by a single GET(SHELF).

What's much more important is that months from now, when coming back to do additional work on the program, the programmer will find it much easier to figure out what is going on amid such complexity. The structuring requirements of Pascal impose organization on the data such that human beings can cope with it better. Just try to imagine how a BASIC puddle of parallel arrays might be constructed to duplicate the comprehensive and versatile BOOK structure illustrated here. In all candor, this BASIC veteran's brain balks at the attempt.

The power of data definition sets Pascal clearly apart from BASIC, and Brown makes his point here with subtlety far superior to that of your reviewer (who tends to come on like 76 trombones).

Pascal doesn't have GOSUB. Instead, the programmer creates PROCEDURES and FUNCTIONS, with names somewhat suggestive of their purpose. These then become new words which the actual program may use as if they were natural-born program words. Put another way, Pascal allows the programmer to build his or her own programming vocabulary, and make the words of the program take on precisely the meaning the programmer desires.

With great civility and British wit, Brown leads his reader to see how all this is done.

Brown acknowledges that BASIC is the best way to write short, quick programs. Pascal's power becomes recognized when long programs are developed by one or more persons over a period of time. Breaking the program into data structures and clearly defined, named procedures and functions allows superior readability.

As an added bonus, Pascal prevents procedures and functions from tromping on one anothers' variables. This allows several programmers to work simultaneously on different parts of a program. The structural requirements standardize the way the parts fit together such that they will mesh with a minimum of trouble. Thus sayeth Brown.

Pascal is not something one can simply read, mark, learn, and inwardly digest. The language wasn't built in a day, and can't be learned in one. But once learned, it has attractions far beyond rude snob appeal.

This reviewer was never able, despite reading seven previous books on the subject, to gain much interest in Pascal. None of the books made the learning effort seem worthwhile. This one did.

Brown's book has that "certain something" we have wanted for more than a year. It is a charming invitation to take up Pascal. You will still need other books to show you how. But this book shows you why.

SIXTH FINGER™

The automatic repeat key for your Apple II*

$9.95

Introductory price

- Adapts your Apple; all keys held down will repeat after the perfect delay.
- Great for program editing, word processing & VisiCalc*
- Easy for anyone to install. Plugs onto the Apple's encoder board (all rev. 7 or later and late rev. 6 with encoder board).

(30 day trial)

Special introductory price	$9.95
add shipping & handling	1.00
Cal. residents add 6% tax	.60

Send check or money order to:

Ivers Specialties
Dept. CA-1
12841 Hawthorne Blvd.
Box 524
Hawthorne, Ca. 90250

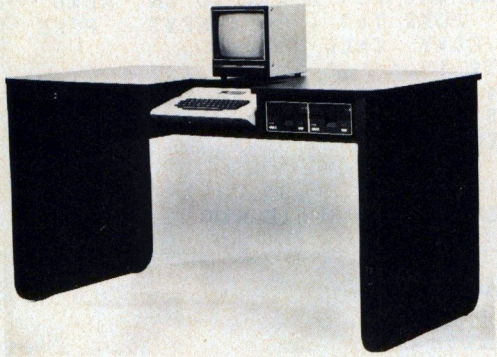
Dealer Inquires Invited

*Apple is a registered trademark of Apple Computer Co.
*VisiCalc is a registered trademark of VisiCorp.

COMPUTER CONSULTING SERVICES

SPACE STATION II

1822 DREW STREET, SUITE 3
CLEARWATER, FL. 33515
(813) 446-2643
(813) 441-3490



This computer work module has been designed specifically for use with the Apple II* computer system. It features:

- 30" x 60" desk height work space
- typing height keyboard channel
- room for two disk drives
- full extension disk storage drawer with lock
- six-outlet power supply
- circuit breaker to protect system from overload*
- 65 CFM super-quiet cooling fan
- complete internal access
- concealed power cords and wiring
- rapid breakdown and assembly for mobility
- clean, contemporary styling with durable laminated finish in a variety of colors and wood grains to suit your decor

Prices*

Space Station II	$395.00
Modesty Panel**	45.00
Monitor Stand**	15.00
Drawer/Shelf Module**	35.00
Castors on Side Panels**	25.00

* Call for quantity prices
** Not shown in photograph

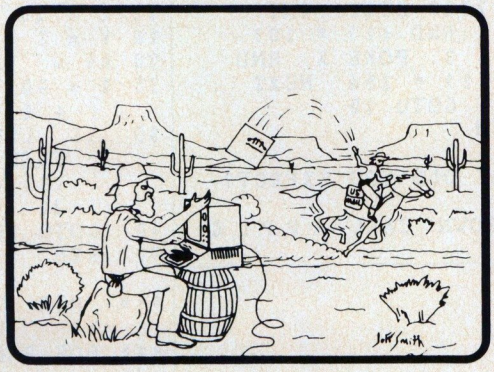
Colors available: solid white, natural almond (to match the Apple)
Wood grains: light oak, winchester walnut
Other colors, wood grains and special real wood veneer finishes available upon request.

Advantages:
- moves disk drives below work surface
- securely locks diskettes away
- removes clutter of disk boxes from desktop
- self-contained, single-switch power supply turns all system components on; one cord from desk to wall outlet
- keyboard is at typing height, work surface is at desk height

*Apple II is a registered trademark of Apple Computer, Inc.

TELEPHONE SOFTWARE CONNECTION™ INC

No MODEM yet??? We'll find another way to reach you! Send for our free catalog.



Software via Teledelivery™!

SHIFT I N # 3 # 3 RETURN
CTRL A CTRL F CTRL A CTRL Q
1-213-516-9432

POST OFFICE BOX 6548 - TORRANCE, CALIFORNIA 90504

© 1982 Telephone Software Connection, Inc.

COMPUTER PRODUCTS
DESIGNING
MANUFACTURING
ELECTRONIC ENGINEERING

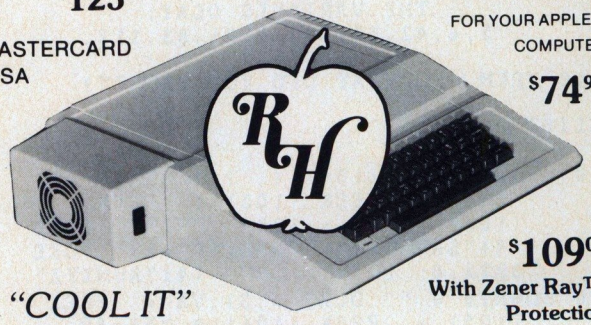
RH ELECTRONICS, INC.
566 Irelan, Buellton, CA 93427
8 to 5 California Time (805) 688-2047

SUPER RAM II™
16K Language Card
With ROM SOCKET
2 Yr. Warranty
$125

MASTERCARD
VISA

COPYRIGHT © 1981 - PATENTS PENDING

SUPER FAN II™
FOR YOUR APPLE II COMPUTER
$74.95



$109.00
With Zener Ray™ Protection

"COOL IT"

- CLIPS ON — NO HOLES OR SCREWS • REPLACEABLE SWITCH
- AVAILABLE IN 120V OR 240V AND 50/60HZ • DURABLE MOTOR
- REDUCES HEAT CAUSED BY EXTRA PLUG-IN CARDS
- INCREASED RELIABILITY — SAVES DOWN TIME AND REPAIR CHARGES
- TAN OR BLACK COLOR • TWO ELECTRIC PLUG OUTLETS
- LOW NOISE DUE TO DRAWING EFFECT OF AIR THROUGH YOUR COMPUTER AND SPECIAL FAN AND MOTOR DESIGN
- "QUIETEST FAN ON THE MARKET"
- TWO EXTRA 120V OUTLETS FOR MONITOR AND ACCESSORIES TURN ON WHEN YOU TURN ON YOUR FAN (NOT AVAILABLE ON 240V MODEL)
- SOLD WORLD WIDE • UNIQUE 1 YEAR WARRANTY

COPYRIGHT © 1981 - PATENTS PENDING

SUPER FAN II™ WITH ZENER RAY™
Transient Voltage Suppressor $109.00
Protect your sensitive electronic equipment.
Voltage spikes can jeopardize your systems.

SO WHAT DID YOU EXPECT DEPT.?

```
]LIST
10  HOME :VT = 8
20  VTAB VT: PRINT "TWO PROGRAMMA
    TICAL PROBLEMS FOR PERUSAL"
30  VTAB VT + 2: HTAB 19: PRINT "OR"
40  VTAB VT + 4: HTAB 8: PRINT "S
    O WHAT DID YOU EXPECT?"
50  IF  PEEK ( - 16384) < 128 THEN 50
60  HOME : POKE  - 16368,0: POKE
    - 21899,160: CALL  - 23186:
    CALL  - 23174
70  REM  ALL PROGRAMS, AND MODS T
    O STANDARD DOS WRITTEN BY BR
    IAN L. MATTHEWS

]LIST

10 : IF  PEEK (104) <  ) 12 THEN
    LOMEM:  PEEK (105) +  PEEK
    (106) * 256 + 1024: PRINT  CHR$
    (4)"SAVEXYZ": POKE 104,12: POKE
    3072,0: PRINT  CHR$ (4)"RUNXYZ"
20 SOWHT$ = "":DLAY = 112:VA =  PEEK
    (105) +  PEEK (106) * 256:AD
    =  PEEK (175) +  PEEK (176)
    * 256 - 26
30  POKE VA + 2,23: POKE VA + 3,A
    D - 256 *  INT (AD / 256): POKE
    VA + 4,AD / 256
40  HOME : POKE 35,25: FOR A = 1 TO
    24: FOR B = 0 TO 1: VTAB A: POKE
    41, PEEK (41) + 4 * B: POKE
    50,255 - 192 * B
50  IF A < 4 OR A > 21 THEN  PRINT
    SPC( 40): GOTO 70
60  PRINT  SPC( A - 4)SOWHT$ SPC(
    21 - A)
70  NEXT : NEXT : POKE 35,24: VTAB 1
80  READ LN: FOR A = 1 TO LN: READ
    B: POKE 767 + A,B: NEXT : POKE
    0,DLAY: CALL 768
90  DATA 19,164,0,136,208,253,173
    ,84,192,164,0,136,208,253,17
    3,85,192,76,0,3
100  REM " SOME GOOD VALUES FOR
        THE VARIABLE DLAY:
      5,6,42,57,64,70,76,83,112,
          130,187
110  REM  SO WHAT DID YOU EXPECT?

]LIST

10 : IF  PEEK (104) <  ) 12 THEN
    LOMEM:  PEEK (105) +  PEEK
    (106) * 256 + 1024: PRINT  CHR$
    (4)"SAVEXYZ": POKE 104,12: POKE
    3072,0: PRINT  CHR$ (4)"RUNXYZ"
20 SOWHT$ = "":DLAY = 31:VA =  PEEK
    (105) +  PEEK (106) * 256:AD
    =  PEEK (175) +  PEEK (176)
    * 256 - 26
30  POKE VA + 2,23: POKE VA + 3,A
    D - 256 *  INT (AD / 256): POKE
    VA + 4,AD / 256
40  HOME : POKE 35,25: FOR A = 1 TO
    24: FOR B = 0 TO 1: VTAB A: POKE
    41, PEEK (41) + 4 * B: POKE
    50,255 - 192 * B
50  IF A < 4 OR A > 21 THEN  PRINT
    SPC( 40): GOTO 70
60  PRINT  SPC( A - 4)SOWHT$ SPC(
    21 - A)
70  NEXT : NEXT : POKE 35,24: VTAB 1
80  READ LN: FOR A = 1 TO LN: READ
    B: POKE 767 + A,B: NEXT : POKE
    0,DLAY: CALL 768
90  DATA 19,165,0,32,168,252,173,
    84,192,165,0,32,168,252,173,
    85,192,76,0,3
100  REM " SOME GOOD VALUES FOR
        THE VARIABLE DLAY:
          21,31,39,55,56
110  REM  SO WHAT DID YOU EXPECT?
```

```
PROGRAM SO_WHAT_DID_YOU_EXPECT;

{ by Tim Jackson }

{ Call -A.P.P.L.E. }

VAR SOWHAT : RECORD CASE INTEGER OF
                1 : (SO : INTEGER);
                2 : (WHAT : ^STRING)
                END;
     DID     : STRING;
     YOU,
     EXPECT : INTEGER;

BEGIN
DID:='SO WHAT DID YOU EXPECT''';
WHILE NOT (LENGTH(DID)=39) DO DID:=CONCAT(DID,'');
FOR YOU:=0 TO 2 DO
  FOR EXPECT:=0 TO 7 DO
    BEGIN
    SOWHAT.SO:=1024+40*YOU+128*EXPECT;
    SOWHAT.WHAT^:=DID
    END
END.
```

```
]LIST

5  GR : POKE  - 16302,0:
     CALL  - 1998
10  FOR X = 1024 TO
      2039 STEP  INT
      ( RND (1) * 10)
      + 3: POKE X, RND
      (1) * 256: NEXT
      : GOTO 10
20  REM

SO WHAT DID YOU EXPECT?

DAVE BROWN : CALL -A.P.P.L.E
```

```
]LIST

10  REM

SO WHAT DID YOU EXPECT?

KEN KASHMAREK : CALL
     -A.P.P.L.E.
12  REM
FIRST KEY IN THIS PROGRAM,
     REMOVE
ALL DISKS FROM DRIVES,
     THEN RUN
100  POKE 512,195: POKE
     513,204: POKE
     514,197: POKE
     515,193: POKE
     516,210: CALL
     65392
```

```
]LIST

10  HIMEM: 16384:
    LOMEM: 8191
12  HOME : HGR
15 X = 20
16 Y = Z = 0
20 A$ =  CHR$ (62)
25 X = 40 *  RND (1)
     + 1
30  IF  LEN (A$) < X
     THEN A$ =  CHR$
     (8) + A$: GOTO
     30
40  GOTO 20
90  REM

***  HGR STRING  ***

*** BOB RAGSDALE ***

* CALL -A.P.P.L.E. *
```

CTRL-C KILLER

S. Scott Zimmerman

I did it in self-defense! I admit that I killed [Ctrl-C], but it was killing me! I had written an innocent Applesoft program which used almost all of the control keys. But many times when I meant to press [Ctrl-X] or [Ctrl-V], I accidentally pressed [Ctrl-C]. My program died on the spot.

My weapon of self-defense is this short assembly language routine Ctrl-C Killer. It is only 42 bytes long, and I stuck it near the top of page three (at $3A2 = decimal 930, above my tone routine at $300 = decimal 768). Killer is completely relocatable.

Ctrl-C Killer is similar to the program ERR-RING written for *Call -A.P.P.L.E.* (February 1981) by Darrell Aldrich, which was suggested to me by Val Golding. The idea behind ERR-RING is to intercept the Apple's output and replace a [Ctrl-G] (bell) with a short tone routine. In Ctrl-C Killer, the keyboard *input* is intercepted and [Ctrl-C] is replaced by a "space" character or some other character of the user's choosing. Actually, Ctrl-C Killer replaces the monitor KEYIN routine, which, among other things, processes keyboard input. For details, consult the Apple] [Monitors Peeled, pages 17-28.

After you type in the listing and assemble it (I use Big Mac), you can BRUN the object program from the keyboard or from within your Applesoft program with PRINT CHR$(4); "BRUN CTRL-C KILLER". Alternatively, you can BLOAD CTRL-C KILLER, then CALL 930. From then on, typing a [Ctrl-C] will usually behave just like pressing the space bar. If you want [Ctrl-C] to act like some other character, just POKE 207,n where n is an ASCII character value 0-255. For example, POKE 207,135 will cause the BELL to sound when [Ctrl-C] is pressed; POKE 207,160 will restore it to "space." POKE 207,131 will essentially disable Ctrl-C Killer since [Ctrl-C] is character 131 ($83). Another way to disable Ctrl-C Killer is to press RESET. To reconnect Ctrl-C Killer, type CALL 930.

Here are a few more notes related to Ctrl-C Killer:

1. ONERR...GOTO will sometimes work to keep [Ctrl-C] from killing your program without having to use Ctrl-C Killer. But ONERR has some problems, and in many cases just doesn't work. With Ctrl-C Killer in the system, *and* with an ONERR...GOTO command, pressing [Ctrl-C] will cause the program to execute the GOTO and give an error #255 (see Applesoft Reference Manual, page 81).
2. Without Ctrl-C Killer, the GET statement will accept [Ctrl-C] without killing the program, but if you press [Ctrl-C] at any other time during execution, you will get bounced out of the program. Ctrl-C Killer will keep you or anyone else from exiting the program at the wrong time.
3. After Ctrl-C Killer is BRUN, a GET or INPUT statement will pick up a "space" (or whatever other character is POKEd into location 207) when [Ctrl-C] is pressed.
4. If you use the syntax

```
100 KB = PEEK (-16384):
    IF KB < 128 THEN 100
```

in your Applesoft program, pressing [Ctrl-C] at that point will usually kill the program. Your self-defense weapon again is Ctrl-C Killer. For best results, BRUN Ctrl-C Killer and then after statement 100 above, place the command

```
110 IF KB = 131 THEN 100
```

Pressing [Ctrl-C] will then have no effect whatever on your program. Without statement 110, KB will *usually* be set to 131 and the program will execute what comes after statement 100. I say *usually* because sometimes the program will not know that [Ctrl-C] has been pressed. In any case, KB will *not* be set by the value at location -16384 ($C000), nor by the value placed in the accumulator after a key is pressed.

5. Ctrl-C Killer will disable [Ctrl-C] even when you *want* to use it, such as aborting a program LIST or getting out of the monitor. Hitting RESET is the only way to abort the listing, and 3D0G or RESET will get you out of the monitor.
6. Ctrl-C Killer will function only with Applesoft INPUT and GET commands. When a keyboard read is used (X = PEEK -16384), program execution will not be interrupted, but X will contain 131 (the negative ASCII value of Ctrl-C). In Integer Basic, the killer prevents aborting a program with Ctrl-C, but the character substituted becomes a "0", rather than a space. If PLE or GPLE are in residence, they will be temporarily disabled by Ctrl-C Killer. PLE or GPLE can be reenabled with & or call 1016, which will also disable Ctrl-C Killer.

So...if [Ctrl-C] is killing your Applesoft programs, kill it first with Ctrl-C Killer. It's all in self-defense.

About the Author

Dr. S. Scott Zimmerman is assistant professor of chemistry at Brigham Young University, Provo, Utah 84602. He is developing computer-based educational software in biochemistry on the Apple to be published by John Wiley & Sons.

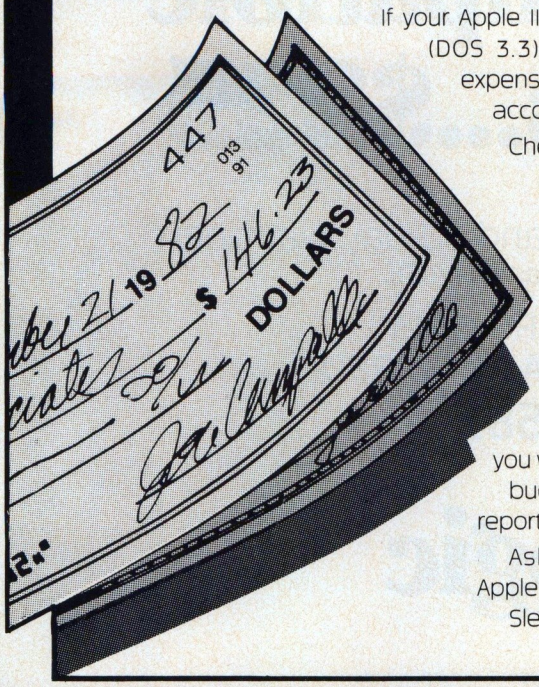
```
]LIST
5  REM
SO WHAT DID YOU EXPECT?
JONATHAN KANDELL : CALL
      -A.P.P.L.E.
10   DEF  FN H(X) = X
20   INPUT "NUMBER 0-254";LI:
       POKE 50, FN H(LI):
     POKE 51, FN H(LI)
30   LIST : GOTO 10
100   REM  SO WHAT DID
      YOU EXPECT?
```

CTRL-C KILLER

```
:ASM

                    ********************************
                    *                              *
                    *        CTRL-C KILLER         *
                    *                              *
                    *    by S. Scott Zimmerman     *
                    *                              *
                    * Call -A.P.P.L.E. : Dec 1982  *
                    *                              *
                    *      This replaces the       *
                    *   monitor KEYIN routine at   *
                    *   $FD1B  (See Apple Ref.     *
                    *   Manual, page 165), and     *
                    *   converts CTRL-C to a       *
                    *   "space" (=$A0), or (with   *
                    *   POKE 207,n) to any other   *
                    *   ASCII number n.            *
                    *                              *
                    ********************************


                          ORG   $3A2        ; = decimal 930

                    BASL    =     $28         ;Screen position
                    KSWL    =     $38         ;Monitor keyboard input switch
                    RNDL    =     $4E         ;Random number
                    CHAR    =     $CF         ;Replacement char (at dec 207)
                    DOSVEC  =     $3EA
                    KEYBD   =     $C000       ;Keyboard input location
                    STROBE  =     $C010       ;Clear input


                    *   Initialization:

03A2: A9 A0               LDA   #$A0        ;Default " " (space) as
03A4: 85 CF               STA   CHAR        ; the replacement char
03A6: A9 B2               LDA   #<CTRLC     ;Set keyboard input vector
03A8: 85 38               STA   KSWL        ;Put it in input switch
03AA: A9 03               LDA   #>CTRLC
03AC: 85 39               STA   KSWL+1      ;HOB of input switch
03AE: 20 EA 03            JSR   DOSVEC      ;Let DOS know
03B1: 60                  RTS


                    *  Keyboard intercept routine:

03B2: E6 4E         CTRLC   INC   RNDL        ;Incr random number
03B4: D0 02               BNE   CTRLC2
03B6: E6 4F               INC   RNDL+1
03B8: 2C 00 C0      CTRLC2  BIT   KEYBD       ;Key pressed?
03BB: 10 F5               BPL   CTRLC       ;No, loop
03BD: 91 28               STA   (BASL),Y    ;Yes, replace flashing screen
03BF: AD 00 C0            LDA   KEYBD       ;Get the pressed key
03C2: C9 83               CMP   #$83        ;Was CTRL-C pressed?
03C4: D0 02               BNE   CONTINUE    ;No, just continue
03C6: A5 CF               LDA   CHAR        ;Act like CHAR was pressed
03C8: 2C 10 C0      CONTINUE BIT  STROBE    ;Clear key strobe
03CB: 60                  RTS

--End assembly--

42 bytes

Errors: 0
```

"Apple II Plus* owners can now have a checkbook package that handles both business and personal checkbook management...CheckMinder offers it."

Rick Taylor, President Sleeping Bear Software



If your Apple II Plus* has 48K and one disk (DOS 3.3), then you can track your expenses with one or more checking account using CheckMinder.

CheckMinder is the easiest to run checkbook management package on the market. It is also extremely flexible, allowing you to track your expenses with up to 999 categories.

The budget management features of CheckMinder give you a variety of reports you will find useful in detailing your budget. If you own a printer, all reports are available for hardcopy.

Ask for CheckMinder at your Apple* dealer or order direct from Sleeping Bear Software.

CHECKMINDER™

The Multi-Checkbook package for Apple II Plus*.

Send check or money order. **$89.95**

Plus $3 shipping and handling
Ohio residents add 5.5% sales tax
(fast delivery)
Dealer Inquiries Invited

*Apple is a trademark of Apple Computer, Inc.
*CheckMinder requires 48K Apple II Plus with one disk drive (DOS 3.3)

SLEEPING BEAR SOFTWARE™
1124 Fuhrman Road
Cincinnati, Ohio 45215
513/769-0692

How to Choose the Best Modem For Your Apple®

Features:	Hayes Micromodem II™	Novation Apple Cat II™	**SSM Apple ModemCard™**
110/300 baud operation	Yes	Yes	Yes
Half/Full Duplex	Yes	Yes	Yes
Auto-Dial/Auto-Answer	Yes	Yes	Yes
Fits completely inside Apple	No	No	Yes
Operates without additional serial interface	Yes	Yes	Yes
Touch-Tone® Dialing	No	Yes	Yes
"Single-Modem-Chip" Reliability	No	No	Yes
Audio Monitor	No	No	Yes
Self Testing	Yes	Yes	Yes
Warranty period	2 yr	1 yr	2 yr
Suggested Retail Price	$379	$389	$299

Trademarks: Micromodem: Hayes Microcomputer Products, Inc. Applecat II: Novation. ModemCard: SSM Microcomputer Products Inc. The Source: Source Telecomputing Corporation, a subsidiary of the Reader s Digest Association. Dow Jones News/Retrieval: Dow Jones & Company, Inc. Apple is a registered trademark of Apple Corporation.

Count the Yes-es. SSM's Apple ModemCard: the most advanced features for the least money. No external equipment required. All other modems are now obsolete.

And with SSM's Transend software, your Apple can send electronic mail or talk to other computers or information services such as the Source℠ and Dow Jones News/Retrieval®. (The ModemCard is also compatible with software for the Micromodem II).

The SSM ModemCard: The only choice for choosy Apple owners. Available from your local computer dealer. Satisfaction is guaranteed. Or your money back.

SSM

SSM Microcomputer Products Inc.
2190 Paragon Drive, San Jose,
CA 95131 (408) 946-7400,
Telex: 171171 SSM SNJ

StarLogic Announces Apple II Compatible Disk Drives...$225

Includes drive, cable, cabinet and 90-day warranty
Also compatible with Franklin ACE

TELEPHONE ORDERS ONLY
MASTERCARD OR VISA REQUIRED
(213) 883-0587

StarLogic

Apple II is a registered trademark of Apple Computer, Inc.

a "HOW TO" guide for users

HARDCORE COMPUTING

About DOS
- How To Write Your Own Data Base
- Using Text Files And EXEC
- Calling DOS Routines from BASIC
- Source Listings of Special DOS Routines

Programming Tips
- Peeks And Pokes
- Applesoft Tricks and Treats
- Ampersand Utilities
- Subroutine Master Library

PLUS: Lots of complete program listings from the HARDCORE program library of copyable software.

NAME _____
ADDRESS _____
CITY _____
STATE _____ ZIP _____

Dept. 2
P.O. Box 44549
Tacoma, WA 98444
(206) 581-6038

$20 U.S.A.
$29 Canada $33 Mexico $42 others
Sample copy: U.S.A. $5 elsewhere $8
U.S. Funds Only No Credit Cards

computer case company

comp case
NOW TOLL FREE
800-848-7548

• AP104

Attaché style cases for carrying and protecting a complete computer set-up. Constructed of the highest quality luggage material with saddle stitching. Will accommodate equipment in a fully operational configuration along with manuals, working papers and disks. Never a need to remove equipment from case. Simply remove lid, connect power and operate.

• AP101	Apple II with Single Disk Drive	$109
• AP102	Apple II with Double Disk Drives	119
• AP103	Apple II, 9 inch Monitor & Double Drives	129
• AP104	Apple *III*, two additional Drives & Silentype	139
• AP105	12 inch monitor plus accessories	99
• P401	Paper Tiger 440/445/460	99
• P402	Centronics 730/737	89
• P403	Epson MX70 or MX80	89
• P404	Epson MX100	99
• P405	IDS 560 or Prism Printer	109
• CC80	Matching Attaché Case (5")	85
• CC90	Matching Attaché Case (3")	75
• CC91	Matching Accessory Case	95

computer case company

5650 INDIAN MOUND CT. COLUMBUS, OHIO 43213 (614) 868-9464

ROBOTICS SERIES

Michael Thyng

2. The Systems Analysis

THIS month I want to go through the steps necessary to define what the robot should do, what it should look like, and what it should be made of. I want to approach the project the way any good Systems Analyst should. That approach is characterized by the following steps:

1. Define the Problem
2. Establish the Scope of the Project
3. Get the User's Approval
4. Design the System
5. Write the Programs
6. Test the Programs
7. Implement the System
8. Get the User's Buyoff
9. Evaluate the System

Remember, this is intended as a learning experience and it will be of greater value if everyone participates. If you see some spot where you think I've missed the boat, be sure to write me.

Defining the problem

In this instance, the problem we are dealing with is defining what we want, what we want it to do, and what we want it to look like. We want a robot. But what kind of a robot? There are industrial, heavy duty robots. There are chess-playing robots that move the pieces for you. There are mechanical arms that hook up to computers that people call robots - but aren't. There are toy robots that "walk" or roll across rooms until they bump into something and stop. There are little wind-up robot-shaped toys that "walk" across tables. There are robots like "Denby" at computer fairs that flirt with young ladies.

My original idea was to build something that did what I told it to do. I figured that if I had complete control of a mechanized object, then I really had a magnificent robot. But in doing my research for this project I came across something that made me change my thinking. I'd like to quote to you from David L. Heiserman's book "How To Design And Build Your Own Custom Robot." In reference to robots as machines, he says, "I have never lost the impression that such machines can, and should be regarded as pets and companions."

That impression colors all my thinking about robots and machine intelligence in general. Unlike so many other experimenters, I have a hard time thinking about these machines as potential slaves that are programmed to do my bidding. I am convinced that thinking about machines as slaves clamps an impenetrable lid on an experimenter's imagination. it seems to straightjacket one's thinking.

"Just play this little game in your head for a couple of weeks. Force yourself to think of robots as pets and companions, instead of slaves. Dream about a machine that can tell you what *IT* wants to do. Censor any thought about your playing a direct role in dictating the machine's behavior. As you work through this exercise, you will find your impression of robots blossoming into new and exciting dimensions. Don't let 'reality' mess up your dreaming! Let yourself go and you'll begin painting a remarkable mental image."

I played that mental game and came away convinced. Now I think my original objective was just a way of making a fancy radio controlled car. I really want to build a robot that will be able to move around my house and tell me what it is discovering. I want it to be able to pick things up and examine them. I want it to get smarter and smarter. And I want to have some way of interacting with it. And, I want to watch it "learn."

The problem then, is to build a working robot that can learn about its environment (both animate and inanimate) and make its own decisions about how it chooses to react to that environment. And I want it to communicate those experiences to me in some form. That robot should look like something I would want to see day after day in my home. It should be made of something that is aesthetically pleasing, yet durable enough to withstand the rigors of bumping furniture.

Establish the goals or limits

Here I want to define the constraints or limits that will govern the project.

1. The robot should be able to move around a flat indoor environment. It must be able to move across both carpet and linoleum.
2. It should be built to move on wheels rather than "legs."
3. It should look like some special little creature that would psychologically endear itself to humans. *(How about "E.T.". . . ed.)*
4. Its internal guidance should be a microprocessor. The microprocessor must be able to receive its program from the Apple][. That way the Apple][can be utilized for programming, and the completed program downloaded to the robot.
5. It should have a radio link to communicate its "thoughts" and experiences to an Apple][computer.
6. There should be no hard connection between the robot and the Apple][.
7. The total cost of the robot project should be less than 700 dollars.
8. The total weight should be less than 25 pounds, with optimum weight at 20 pounds. That way, it can be moved around more easily. The less it weighs, the less energy it requires between recharges.
9. It should have an arm capable of lifting a minimum of 1 lb.
10. It should be able to tell when its power is getting low and seek a source of replenishment.
11. It should have enough on-board sensors to keep itself from harm.
12. It should have sensors of sufficient quality to distinguish different shapes of approximately the size of a pop can or cup.

—◉→

13. It should be goal oriented on multiple levels. For example, On the most basic level, humans have the goal of continued breathing. On a higher (or lower) level, we have a goal or desire to eat. On a still higher level we have a desire for knowledge. And on it goes. Now I may have stated the concept incorrectly, but I think you can understand the idea I'm trying to express here.
14. It should have a personality of its own.
15. It should be able to run a minimum of 2 hours between recharges.
16. Aluminum should be used wherever feasible. It is abundantly available, and has an acceptable strength to weight ratio.

Get the user's approval

Since I am the primary user and designed the project, I essentially have given the project "go-ahead" approval.

Design the system

This is a toughie. If I was an engineer with a substantial budget, I would specify the kinds of parts I chose to use, the exact design to which this project would conform, and the kind and capability of microprocessor that "had" to be used. Unfortunately for uniformity, that just isn't the case. I don't know what kind of motors will be used because they will be purchased based on a price/performance trade-off. I don't know what kind of batteries, microprocessor, wheels, or specific sensors will be used either for the same reason.

What then can be designed? The general size, shape, and type of locomotion can be designed now, with the understanding that all dimensions are negotiable - not fixed. I want to have a finished robot that will be large enough to be seen in time to avoid bumping it accidently while walking through the house. Making it a little on the large size would be an advantage for any bulky parts we may try to put in. There would be some natural size limitations imposed. One is the size of the average doorway. The other is the power consumption/weight trade-off. Based on what I have read, Lead-acid batteries are the only practical power source for this price range and robot size. I think I'll use separate batteries for motive and circuit power.

Next time - where to go for materials.

If you are interested in contributing ideas to this project, WRITE to me at:

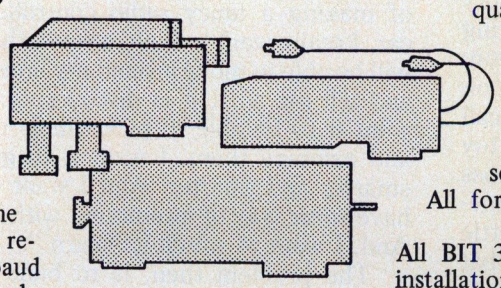
Michael Thyng
Robotics Project
6242 - 35th N.E.
Seattle, WA 98115

I will answer all your reasonable questions as quickly as possible either by return mail or through this column.

apple add-ons

APPLE ADD-ONS are products by BIT 3 Computer Corporation to increase the capability of your APPLE II or APPLE II PLUS. Make it an 80-column word processor with the FULL-VIEW 80 . . . or a full-fledged terminal with the DUAL-COMM PLUS . . . or increase its memory with the MEMORY PLUS.

DUAL-COMM PLUS
adds two fully independent serial I/O interfaces to your APPLE II. Terminal mode firmware converts an APPLE into a full or half duplex computer terminal. With a modem such as the NOVATION CAT™ or the HAYES SMART-MODEM™ connected to one serial port and a printer connected to the other port you can use the APPLE as a remote terminal with online printer. 16 baud rates are available from 50 to 19200 baud. Price: $239.00



The FULL-VIEW 80 provides an 80 x 24 display yet retains the standard APPLE display too. You can switch between 80 and 40 column mode via keyboard or program. An 8 x 10 cell size character set provides quality characters, full descenders, and line-drawing graphics. The

FULL-VIEW 80
is compatible with APPLE peripheral cards, Basic, Pascal and several word processing systems. All for $295.00 – no extras needed.

All BIT 3 products feature easy plug-in installation and are designed with you, the user in mind. Contact your local APPLE dealer or our offices for more information.

The MEMORY PLUS
adds 16K of RAM memory for your 48K APPLE II or APPLE II PLUS. The addition of the MEMORY PLUS represents a 33% capacity increase in RAM memory and 40% more available RAM for disk users since APPLE DOS occupies almost 11K of memory. It is compatible with Integer Basic, Applesoft Basic, Pascal, Z80 Softcard, Fortran and other products such as Lisa 2.0 and Visicalc 3.3. Price: $149.00

BIT 3
COMPUTER CORPORATION
8120 PENN AVENUE SOUTH, SUITE 548
MINNEAPOLIS, MN 55431
(612) 881-6955

Apple II is a registered trademark of Apple Computer, Inc.; Z80 Softcard is a registered trademark of Microsoft Consumer Products. D.C. Hayes Smart-Modem is a trademark of D.C. Hayes, Associates, Inc.; Novation CAT is a registered trademark of Novation, Inc.

MasterCard VISA

EAMON ADVENTURES:

OUR BIGGEST CONTEST YET!

BEGINNING now, we are accepting entries for our EAMON ADVENTURE Contest. Don Brown, our September A.P.P.L.E. meeting guest, wrote the original EAMON ADVENTURE game. He and other creative people have written many adventures since then. There are currently about 30 adventures, all public domain software and all available from A.P.P.L.E.

Your mission, if you choose to accept it, is to write a new EAMON ADVENTURE.

The Rules:

1. All A.P.P.L.E. members are eligible to enter except EAMON creator, Don Brown and A.P.P.L.E. employees.
2. Don Brown will judge the entries and select the first, second, and third place winners. His decision will be final.
3. All adventures submitted become property of A.P.P.L.E.
4. To have your adventure eligible for judging, it must be within the bounds of good taste. Use your own best judgement. Observe any of the currently available adventures for additional guidance. It should be in English. The decision of the Contest Coordinator or the Judge shall be final.
5. No entries will be returned.
6. All adventures deemed eligible for judging will become public domain software. A.P.P.L.E. will make copies of them available for purchase after the contest.
7. Your adventure must conform to the format established by Don Brown. You can see the format on any of the EAMON ADVENTURES which are available from A.P.P.L.E.'s "As-is" software. On Volume 144, Don Brown has set up an adventure creation disk—which also contains THE BEGINNERS CAVE.
8. All adventures must be winnable. That is, if it is impossible for a character to ever make it through your adventure without being killed, then your adventure does not meet the requirements for eligibility. Note, you don't have to be able to win every time, it just has to be possible with some regular characters.
9. Hints: Don Brown does appreciate puns. Consider a theme for your adventure and carry that through your adventure. The EAMON ADVENTURE called Death Star was a Star Wars take-off. You might consider something that is happening or has happened in the news. Or consider Politics, Sports, Education, the Arts, the Computer Industry, a TV program, something based on your job, the future, the past, or something funny, scarey, or intellectual.
10. The deadline. All program diskettes must arrive by Midnight, Monday, February 28th, 1983 and will not be returned.
11. All entries must be on 5 1/4 inch floppy diskettes.
12. Mail one or two diskettes to the contest director. If the disk drives that the Judge or Contest Director use cannot read your diskette, your entry will be rejected. You might consider mailing two diskettes to help ensure that we can read your entry.
13. All contest adventures must be able to be run after the Beginners Cave, just as all other adventures are.
14. We expect between 20 and 100 adventures to be entered in the contest. Consequently, it may be quite some time after February 28th before we know who the winner is. Because of that we can't tell you exactly when the winner will be announced. However, the Contest Coordinator will tell you how many entries were received and how many were deemed eligible shortly after February 28th.
15. You may enter as many different adventures as you wish.
16. Mail your entries to:

    EAMON Contest Coordinator
    c/o A.P.P.L.E.
    304 Main Ave. S.
    Renton, WA 98055

17. If you have questions, you may call the Contest Coordinator at (206) 524-2744. Pacific Time Zone 9AM to 3PM. (Don't just call to say you are writing an adventure.)
18. This contest is void where prohibited by law.

The PRIZES:

First prize - a $250 credit for any product available through A.P.P.L.E.

Second prize - a $100 credit for software or publications available through A.P.P.L.E.

Third prize - a $50 credit for software available through A.P.P.L.E.

In one respect everybody wins. The number of EAMON ADVENTURES available will grow. Start today.

```
]LIST

10  REM

SO WHAT DID YOU EXPECT

VAL J GOLDING

CALL -A.P.P.L.E.

100 HEX$ = "0:A0 3 98
      N303:43 41 4C
      4C 20 2D 41 2E
      50 2E 50 2E 4C
      2E 45 2E 22 0
      ND9C6G": FOR I
      = 1 TO  LEN (HEX$):
      POKE 511 + I,
      ASC ( MID$ (HEX$,I,1))
       + 128: NEXT :
      POKE 72,0: CALL
       - 144
110  CALL  - APPLE
```

SO WHAT DID YOU EXPECT?

OUR SMALLEST CONTEST YET!

Help Wanted:

FIVE judges for our So What Did You Expect contest as announced in the August issue of *Call — A.P.P.L.E.* Judges will be selected by lot from all applications postmarked prior to Feb. 1, 1983, and must include the following information with their application to judge. A.P.P.L.E. employees are not eligible to participate.

1. Name
2. Address
3. Phone
4. A.P.P.L.E. membership number

Those judges who submit final scoring on So What will receive their choice of any A.P.P.L.E. software package. All communications shall be addressed to "So What Contest" *Call — A.P.P.L.E.* Editorial Department, 5606 California Ave SW, Seattle, WA 98136. Potential submitters of So What Did You Expect for 1983, when we **expect** to run a similar contest, would be well advised to read the following scoring criteria upon which the judges will base their points.

Scoring Criteria

Four categories have been established that points may be awarded on. Listed in order of ascending importance and with point ranges, they are:

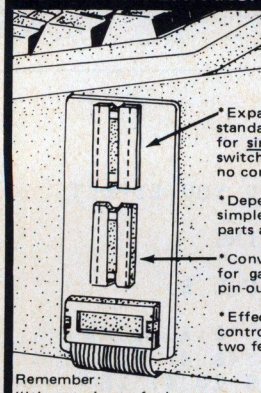
1. Originality of idea (0—10)
2. "Cute" or humorous aspects (0—20)
3. Uniqueness of application. Has it been done before? Does it introduce new or altered programming concepts? (0–30)
4. When the program is run, does it cause the reader to delve into it, to find out what makes it tick, to make the reader think, and most importantly, to learn? (0—40)

Prizes will be awarded on the basis of averaging scores calculated by the five judges. In addition, any So What entry which does not otherwise qualify as a winner, based on the composite scoring, but which was scored first by any individual judge, will receive an appropriate supplemental prize.

Each judge will receive photocopies, appropriately numbered, of all So What submissions. Those numbers will be the sole reference used for point scoring. All So What's published in 1982 will be eligible for consideration, including "All About DOS," irrespective of publication date.

ProLine X-10

EXTERNALLY ACCESSABLE GAME PORT WITH EXPANSION FOR APPLES.



*Expansion socket accepts standard game controllers for simultaneous use. . .no switches. . . no jumpers. . . no confusion!

*Dependable, easy to install, simple, inexpensive, quality parts and construction!

*Convenient external socket for game I/O port with all pin-outs present!

*Effectively increases game controller cable length by two feet!

Remember:
"It's not nice to fool with the mother board!"

See your dealer or order direct. . .Add $1.00 handling on direct orders. Check, Visa or Master Card is acceptable.

ProLine X-10 Only $19.95

Pro Line Electronics

Box 735, Higginsville, MO 64037

Apple is a registered trademark of Apple Computers, Inc.

THE WALL STREET PLOTTER

For use with Apple™ Computer Systems

This program can help you make money, and it's so easy. Two separate price move indicators may be displayed with the standard high-low-close financial plot, the popular moving average, or a trend analysis of the trading cycle. The trend analysis overlays the trading cycle onto buy and sell zones resulting in a practical, easy to use price move indicator. Plots may be generated for any financial security, commodity, or market average.

This package is compatible with text files containing historical quotes captured from several of the popular quote services by any of the popular telecommunications packages. The package also includes a powerful line oriented text editor for manual data entries, deletions, or additions. Data Files may be created in whole or in part with the editor.

The package is compatible with all of the popular graphics printer and plotter drivers as well as our other product, The Superplotter.

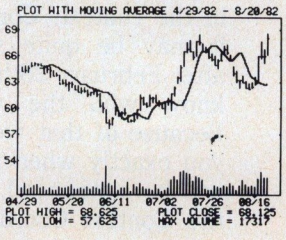
Requirements: 48K Apple™ II Plus with DOS 3.3
or Apple™ III in emmulation mode

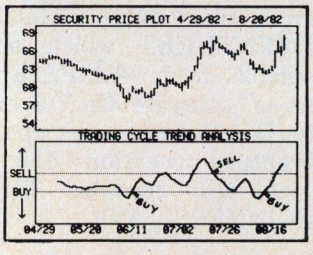
Price $125.00

See your local dealer for a demonstration or order direct from:
478 Engle Dr. / Tucker, GA 30084 / (404) 491-7905
Outside Georgia: (800) 241-6753 ext. 503

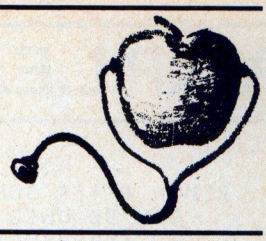
VISA & MasterCard accepted. Mail Orders: Add $2.50 shipping and handling ($5.00 outside USA)

Dickens Data Systems





A.P.P.L.E. DOCTOR



Cornelis Bongers/Call -A.P.P.L.E. Staff Writer

Question:

Would you or one of your staff writers please explain why the two rounding programs do not produced the same results.

```
]LIST

100  INPUT X
110 X =  INT ((X + .005)
      * 100) / 100
120  PRINT X

]RUN
?4.005
4
```

```
]LIST

100  INPUT X
105  LET Y = (X + .005)
      * 100
110 X =  INT (Y) / 100
120  PRINT X

]RUN
?4.005
4.01
```

Answer

The problem you describe is a common one and it is caused by the fact that not all FP numbers can be represented in a finite expansion. What this means is illustrated by the following example:

3 x 1/3 = 1

but

3 x .33333 <> 1

If you try to convert 1/3 to the decimal system, then no matter how many digits precision you take a multiplication with 3 never leads to the correct answer (i.e., 1).

Likewise, if you calculate the expression

(X+.005) x 100

the result is slightly less than 401 if X =4.005. As you may know, results of calculations are stored in the main floating point accumulator (MFP). The MFP uses for most calculations not 4 bytes for the mantisse, but 5. When the contents of the MFP have to be stored in a variable (i.e., Y=(X+.005) x 100), the mantisse of the MFP is rounded off to 4 bytes, being the number of bytes available (for the mantisse) in the storage area of real variables. The rounding is done according to the value of the 5th mantisse byte of the MFP.

The result of the expression (1) equals

89	C8 7F FF FF	D0
exp	mantisse	5th byte of m.*

(*approx 400.9999...)

so rounding off leads to

89 C8 80 00 00

which is the correct value of 401.

Hence, Y in line 105 of program number 2 becomes 401 and X assumes the value 4.01 after execution of line 110.

However, in program number 1, X is calculated by a one liner. Applesoft first calculates the part of the expression that corresponds to (1) above. Next the INT function is applied to the MFP, but since the MFP contains approximately the value 400.9999.., the result will be 400. As a consequence X assumes the value 4.

In summary, the cause of your problem is thus that the MFP is not rounded off (to a 4 byte mantisse) before the INT function is taken. This is a decision made by the designers of Applesoft and it is not unreasonable, for rounding off the MFP before taking (f.i.) the INT function may lead to the result:

INT ((4.005+.00499999999)×100)/100 = 4.01

which is equally undesirable.

As a general rule, you should keep in mind that when working with FP values, results need not be exact. By the way, if you are bothered by output problems in Applesoft, why not use & PRINT USNG "0.00";X (from "Print Using and Friends" *(Call -A.P.P.L.E. in Depth 1, All About Applesoft)* which is more convenient and reliable than doing the rounding off in BASIC.

For a different approach and additional explanation, the reader is directed to David Lingwood's article in this issue on problem solving.

Question

I have enjoyed immensely your articles in Call - A.P.P.L.E. *and have found them to be very informative. I have a question that I would appreciate your help on. The question is how can I PEEK/POKE the drive number in DOS? I have had no luck at figuring out the answer to this query.*

Question

I have been using your DOS MOVER from Call -A.P.P.L.E. *with great success. However, prior to DOS MOVER, I saved much time by turning on the disk drive motors during long reads and writes, then off again.*

I would appreciate it greatly if you could supply the correct addresses now that DOS is moved. I have tried unsuccessfully.

Answer

The listing shown on page 80 shows how to PEEK/POKE the drive number in DOS. For this purpose a small ML program is POKEed at $300 first. In case other routines reside at $300, you have to POKE the routine elsewhere (and adjust the CALL instructions appropriately), which causes no problems as the routine is relocatable. A special note, it is best to try this on a scratch disk first, since one typo could wipe out an entire disk.

Question

I want to know why some commercially available programs such as Apple's CONTROLLER and A.P.P.L.E.'s APMAIL choose to disconnect DOS when activating the printer. They take the trouble to PEEK the I/O hooks and save them, then use PR#0 followed by PR#1. Later they have to POKE the DOS vectors back in the hooks. What does DOS do to the printer and whatever happened to CALL 1002?

Answer

Both the CONTROLLER and APMAIL go back to the bugs of DOS 3.1. The reason for resetting the hooks (which temporarily disabled DOS) was to prevent keyboard entry from erroneously being misinterpreted as a DOS command. To illustrate, an INPUT statement which requested entry of a last name (in this example, the name is BRUNGAARD) would have been misinterpreted by DOS 3.1 as a command to BRUN a file named GAARD, and would have created an error condition when the file was not found, and would also write a file title of GAARD to the directory. In addition, 3.1 DOS did not have the $3EA vector (CALL 1002) to reconnect DOS.

A.P.P.L.E. DOCTOR

```
]LIST

1   REM    ***** SOME ROUTINES THAT
       CAN BE USED WITH DOS MOVER
       AND AMPERSOFT
2   REM   ***** CALL 768:PRINT PEEK
       (255) RETURNS CURRENT DRIVE
       #
3   REM   ***** POKE 255,2:CALL 768
       +15 SETS CURRENT DRIVE # TO
       2
4   REM   ***** CALL 768+30 LET MOT
       OR STAY ON AFTER DISK OPERAT
       ION
5   REM   ***** CALL 768+45 TURNS M
       OTOR OFF
7   GOSUB 1000 : REM  POKE ML PROG
       RAM AT $300
10  GOSUB 100: REM  PRINT CURRENT
       DRIVE NO
20  PRINT : INPUT "ENTER DRIVE #
       ";B: REM  NOW ASK FOR NEW DR
       IVE #
30  POKE 255,B: REM  STORE DRIVE
       # IN LOC $FF
40  CALL 768 + 15: REM  STORE DRI
       VE # IN DOS
50  GOSUB 100
60  CALL 768 + 30: REM   DISABLE
       DOS INSTRUCTION TO PUT MOTOR
       OFF
65  PRINT : PRINT "MOTOR WILL STA
       Y ON AFTER NEXT CATALOG"
66  FOR I = 1 TO 1000: NEXT
70  GOSUB 100
75  PRINT : PRINT "MOTOR STAYS ON
       FOR A WHILE"
80  FOR I = 1 TO 4000: NEXT : REM
       WAIT SOME TIME
90  GOSUB 100
95  CALL 768 + 45: REM  TURN MOTO
       R OFF (AND RESTORE DOS INSTR
       UCTION)
97  PRINT : PRINT "END OF DEMO"
99  END
100  REM  PRINT CURRENT DRIVE # A
       ND CATALOG
110  CALL 768:A =  PEEK (255): PRINT
       "DRIVE ";A;" IS ACTIVE"
120  PRINT  CHR$ (4);"CATALOG"
130  RETURN
990  REM  ***** WARNING *****
995  REM    BE SURE YOU COPY THE
       VALUES BELOW EXACTLY, FOR
996  REM  ONE ERROR MAY DESTROY A
       DISK IN A SECOND.
997  REM  PREFERABLY THE ROUTINE
       SHOULD BE TESTED
998  REM  WITH A SCRATCH DISKETTE
       FIRST.
1000  DATA  173,131,192,173,131,1
      92,173,104,234,133,255
1010  DATA  173,130,192,96,173,13
      1,192,173,131,192,165,255
1020  DATA  141,104,234,173,130,1
      92,96,173,131,192,173
1030  DATA  131,192,169,96,141,77
      ,254,173,130,192,96,173
1040  DATA  131,192,173,131,192,1
      69,189,141,77,254,141,232
1050  DATA  192,173,130,192,96
2000  FOR I = 768 TO 830: READ A:
       POKE I,A: NEXT : RETURN
```

```
]LIST

5  REM

SO WHAT DID YOU EXPECT?

BILL BOUMA : CALL -A.P.P.L.E


7  ONERR  GOTO 100
10  DIM ST(32,1):SP
       = 0:LN = 120:P4
       = .785398163:AN
       = SP:S2 =  SQR
       (2)
20  HGR2 : HCOLOR= 7:X
       = 79:Y = 60: HPLOT
       X,Y
30  IF LN > 2 THEN 50
40 X = X +  COS (AN
       * P4) * LN:Y =
       Y +  SIN (AN *
       P4) * LN: HPLOT
        TO X,Y:SP = SP
       - 1:LN = ST(SP,0):AN
       = ST(SP,1): GOTO
       30
50 LN = LN / S2:ST(SP,0)
       = LN:ST(SP,1)
       = AN - 1:AN =
       AN + 1:SP = SP
       + 1: GOTO 30
100  END
```


BEEP! LOST YOUR PROGRAM?

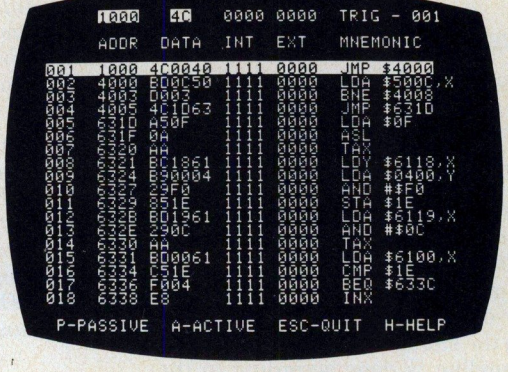
BUS RIDER
LOGIC ANALYZER FOR THE APPLE II

The Bus Rider is a self diagnostic development tool that allows real time analysis of software and hardware in the Apple II computer.

The Bus Rider provides:

- Monitors and saves 512 cycles of the address and data bus, NMI, IRQ, DMA, R/W and 4 external lines.
- Pretrigger viewing of up to 512 samples.
- 4 external inputs with variable threshold reference.
- Display cycle by cycle execution or 6502 disassembled code.

The Bus Rider comes complete with Bus Rider circuit card, reference manual, Bus Rider software diskette, and 10 easy hook external input cable. The total system price is **$395.00**



Bus Rider – Disassembled Display

RC Electronics Inc.
5386 Hollister Avenue, #D
Santa Barbara, CA 93111
(805) 968-6614 • TELEX 295281

MC Visa

The arcade-warp is open!

YOU CAN RUN BUT CAN'T HIDE. YOU CAN KILL BUT CAN'T ESCAPE.

Tubeway! It's an insidious invasion route created by beings from a parallel universe—a strange, geometric universe. You're trapped on the rim as their fleet swarms out of the warp on a voyage of conquest. The battle is yours alone—and it's far from easy because normal strategy doesn't work! . . . you have to fight by their strange, geometric rules!

Here's the fastest, most fascinating of the new style space games. So involving and exciting it's destined to become an all-star, all-time hit. Be one of the first to take on the challenge of the lightfast Tubeway!

$34.95 for the Apple II*. At your computer store or:

TUBEWAY

DATAMOST INC

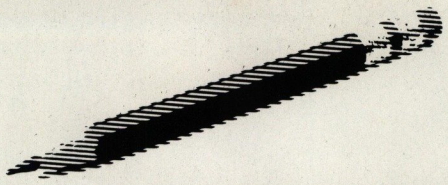
9748 Cozycroft Ave., Chatsworth, CA 91311.
(213) 709-1202

VISA/MASTERCARD accepted. $2.00 shipping/handling charge. (California residents add 6½% sales tax.)

COREY WOLFE

*Apple II is a trademark of Apple Computer, Inc.

WRITE -A.P.P.L.E.



The Readers

Splitting Up

Dear Val,

I recently had need to use Charles Kluepfels excellent SPLITTER routine. He notes in the October 1980 issue (as well as the first In Depth volume) that after splitting a program that it can be re-entered only with a RUN command. Indeed, after activating the routine with the necessary POKEs to $6 (the length of the gap) and $7 (starting location of the gap), and CALLing 37888 from within a program, the program is aborted and the user placed squarely in Applesoft. This can be avoided by substistituting the following bytes in the SPLITTER routine itself:

949F: 12 D9

Calling the modified SPLITTER will allow the Applesoft program to continue uninterrupted.

Bruce McCoy
163 Union City Rd
Prospect, CT 06712

A TUTORIAL on using the Splitter is forthcoming shortly, along with a reprint of code for the benefit of those readers unfamiliar with it.

The Source

Dear Val,

I saw in the August *Call-A.P.P.L.E.* that your club is offering special prices for members for THE SOURCE....My club is negotiating a similar sort of arrangement with a local look-alike called "The Australian Beginnings." ...(also) we hope to use a bulletin board and the mailing system for club information.

Could you pass this letter along to your SOURCE experts with a request for any advice they may care to give us. We would be most grateful to hear of any pitfalls before we discover them the hard way....

Denis Kennedy, President
Victorian Apple Computers in Education
P.O. Box 108
Broadford, Vic. 3658
Australia

HOW ABOUT IT? If you've experience and tips in setting up a club communication system, SOURCE, or otherwise, drop Denis a line. Send us a copy, too; several folks are interested in the experiences of network builders.

New Technique

Dear Val,

Some of your newer users may not know they can easily draw horizontal and vertical lines on their screens using the INVERSE command instead of graphics mode.

Example using lines to divide text material:

LISTING 1

```
]LIST

5   IF  PEEK (978) =
      190 THEN 140
7   REM  LINES
10   HOME
15   REM  PRINT HORIZONTAL
       LINE
20   FOR C = 1 TO 20
30   INVERSE
40   VTAB 14: HTAB C:
       PRINT " "
50   NEXT C
55   REM  PRINT VERTICAL
       LINE
60   FOR R = 8 TO 20
70   HTAB 20: VTAB R:
       PRINT " "
80   NEXT R
90   NORMAL
```

Example drawing a horizontal line from a variable:

LISTING 2

```
]LIST

5   IF  PEEK (978) =
      190 THEN 140
7   REM  VARIABLE LINE
10   REM  SCALE OF 1
       COLUMN =5%
15   REM  PRINT HORIZONTAL
       LINE
20   FOR C = 1 TO 20
30   HOME
40   VTAB 9: HTAB 1:
       PRINT "NAME X"
50 P = 65: REM  SUPPOSE
       65% HAS BEEN CALCULATE
       D FOR NAME X USING
       VARIABLE"P"
55   REM  PRINT VERTICAL
       LINE
60 CP = 10 + P / 5:
       REM  START IN
       COL 10+(%/SCALE
       OF 5% PER COL)
70   FOR C = 10 TO CP
80   INVERSE
90   VTAB 9: HTAB C:
       PRINT " "
100   NEXT C
110   NORMAL
120   VTAB 9: HTAB (CP):
       PRINT P"%"
```

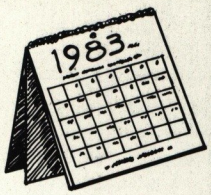
To add another % line, change the VTAB number, name, and value of variable.

B. J. Wilson
P.O. Box 12628
Seattle, WA 98111

PS: Thank you for an informative magazine, I have been told it is not posible to make horizontal lines in text material on an Apple.

```
]LIST

5   REM

SO WHAT DID YOU EXPECT
      DEPT

ALAN BAXTER-CALL -A.P.P.L.E.


10   DATA  67,65,76,76,45,65,
       46,80,46,80,46,76,46,6
       9,46,33,32
20   TEXT : HOME :XO
       = 18:YO = 15:A
       = 3.5:DANG = 0.58905:A
       NG = 9.42478:I
       = 1
30   READ TITL:X1 =
       COS (ANG) * A
       *  SQR (ANG) +
       0.5:Y1 =  SIN
       (ANG) * A *  SQR
       (ANG) + 0.5
40 X = XO +  INT (X1):Y
       = YO -  INT (Y1):
       VTAB Y: HTAB X:
       PRINT  CHR$ (TITL)
50 K =  PEEK ( - 16384):
       IF K > 197 THEN
       90
60 ANG = ANG - DANG:
       IF ANG < 0.5 THEN
       ANG = 9.42478:
       POKE 1849,106
70 I = I + 1: IF I >
       17 THEN I = 1:
       RESTORE
80   FOR J = 1 TO 200:
       NEXT : GOTO 30
90   POKE  - 16368,0:
       NORMAL : END
100   REM

IF YOU GET BORED, CHANGE
A TO 1.6 IN LINE 20
AND SQR(ANG) TO
ANG IN LINE 30
```

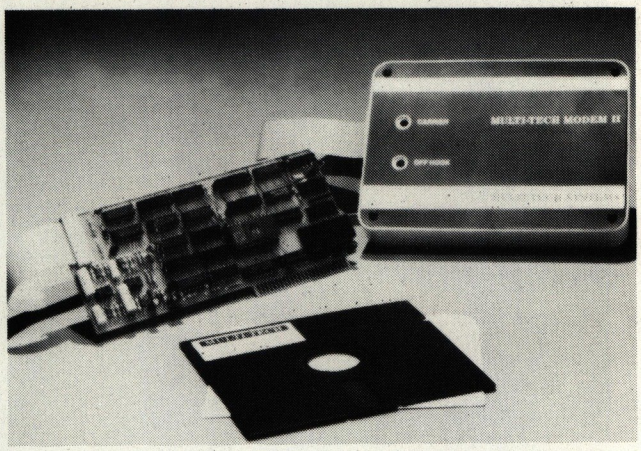
TOMORROW'S APPLES TODAY



Kevin O. Donohoe/Editor

NEW USER-FRIENDLY MODEM AVAILABLE FOR APPLE II AND APPLE II PLUS

Designed by Multi-Tech Systems, Inc. for use with an Apple II, Apple II Plus or Bell and Howell computer, the user-friendly Modem II features complete menu driven software with user prompts at all levels of command entry and keyboard dialing for ease of operation.



The Modem II does not require the use of a serial interface card. It plugs directly into one of the I/O slots inside the computer for full or half duplex communications at either 110 or 300 bits per second. Software compatible with other Apple II smart modems, this unit provides crystal-controlled, digital transmission and reception of data to other Bell 103-type equipment with a high degree of accuracy.

It is equipped with firmware for direct communications from a "bare bones" computer and comes with a software disk containing various utility programs including the terminal program. This FCC registered unit is also equipped with a built-in speaker to allow audible monitoring of call progress as well as auto-dail and auto-answer. The suggested retail price for the unit is $369.00.

Multi-Tech Systems, Inc., 82 - Second Avenue SE, New Brighton, MN 55115, (612) 631-3550.

MAYDAY™

The **MAYDAY 60** is designed especially for those small microcomputers that must have a true 60 cycle sine wave power to operate AC motors, fans, etc.

MAYDAY 60 is a switchable Uninterruptible Power Supply developed by Sun Research, Inc. to provide complete brownout and blackout protection for mini/microcomputers and microprocessors or electronic equipment.

Our **MAYDAY 60** systems are designed to give the operator a minimum of 30 minutes of user time at the rated wattage of the system with a fully charged battery. **MAYDAY 60** maintains the program in the computer and also allows the operator to save the program of data on a disk by performing a soft shut down.

MAYDAY 60 provides a two-stage regulated battery charger to insure that the battery is always ready to be used. Available accessories include our Constant Voltage Regulator when using normal A.C. power (see our MDS Series), a smart circuit that provides battery condition indicator lights, low batter alarm and computer I/O port for status reporting.

The **MAYDAY 60** series comes in three wattages: 150, 250, and 600 watts. The "S" designation indicates that this model is equipped with the Line Surge Protector (see MDS series). Also see specifications on the reverse of this sheet.

All **MAYDAY 60** series UPS systems come complete with battery pack, which includes a 12-volt Deep Cycle Battery in a plastic marine battery case. Typical recharge time is 4 to 6 hours for 90% charge replacement. Optional battery packs available are a non-liquid lead acid battery or a sealed cell pack. These may be obtained locally or from Sun Research, Inc.

MAYDAY 60 comes complete and ready to be plugged in. No rewiring of your unit is necessary. Some mini/microcomputers include all Radio Shack Model II and 16, Commadore, Ohio Scientific, IBM, Apple, etc., including hard disk drives.

01

MAYDAY	150watts	$ 400.00
MAYDAY60+S	150watts	475.00
MAYDAY60+2	250watts	525.00
MAYDAY60+2S	250watts	700.00
MAYDAY60+6	600watts	900.00
MAYDAY60+6S	600watts	1140.00
MAYDAY60+10*	1000watts	1950.00

*Switch-over time 50 ms

"S" designator denotes surge protection built in.

BASIC EXTENSION

Don't waste time waiting for the computer! Get the job done fast! By using Basic Extension the efficiency of your programs can be substantially improved, and the development time reduced. No knowledge of machine language is required. All you need to know is basic!

Basic Extension is a group of machine language routines (invokable moduals) that can be added into your Business Basic program. There are three types of routines: disk, array, and utility. The disk routines can speed up disk access by a factor of ten, save 20%-30% on disk space, and allow access to files of any type. Array routines dramatically spped up the manipulation of string and numeric arrays (great for basic sorts and insertions). The utility routines add features that are not normally utiltiy routines add features that are not normally available in Business Basic: reset disable, string upshift, typehead buffer set/clear, and more!

Speed up your programs, save disk space, add new functions, decrease development time, ideal for software developers and independent programmers.

Basic Extension is a handsomely bound two-disk set containing invokable moduals, extensive documentation and demonstration programs.

Note: Also included as a demonstration program, is a 'font' file editor. It will allow you to easily edit the character sets stored in font files. Requires Apple /// Business Basic. Suggested retail: $95.

How to get your fir

You're not alone, you know.

We've all been through that same excruciating moment when we realized that we were going to have to learn a lot more than we wanted to know about programming, in self-defense.

But most of us survived.

Many of us with a little bit of help called dBASE II, the relational database management system (DBMS) for micros.

The best defense is a strong offense.

Don't get seduced by BASIC, because basically BASIC is weak. And to make it do anything useful can take 10 times as much programming (and time) as with dBASE II.

dBASE II is much easier because it's a relational DBMS, and you control your information with powerful, English-like commands.

With a word or two, you CREATE databases, APPEND new data, UPDATE, MODIFY and REPLACE fields, records and entire databases. DISPLAY some or all of your data for any conditions you want to apply. Organize months worth of data in minutes with REPORT.

You can prepare your forms and formats precisely the way you want them. Do calculations on fields, records and entire databases with 10-place accuracy.

And do even more with dozens of other commands.

Here's the catch.

With any language, you'll need to understand relational operators ("less than," "greater than," "equals") and a bit of logic ("or," "and," "not").

With dBASE II, you'll also get an easy way to make choices (IF..THEN..ELSE) and a powerful way to perform repetitive tasks (DO WHILE..).

With these tools, you're ready to tackle your accounting and time billing, project management, and any other data handling and record keeping. You can work interactively and get your answers right now. Or save your instructions and repeat everything with two words: DO Manhours, DO ProjectX, DO whatever has to be done.

Use dBASE II to help make your choice.

If you've got a 48k CP/M micro with a disk drive (96k IBM PC), send us $700 ($400 for a 56k Apple). We'll send you a copy of dBASE II to use free for 30 days.

the best of st micro.

Instead of just poring over a manual, run it and make certain that dBASE II does what you need done.

Then if you find it isn't right for you, send it back and we'll return your money, no questions asked.

But we know that you'll probably keep it.

Because having dBASE II is like having a black belt in micros.

Call (213) 204-5570 today or drop by your local computer store for the rest of the story.

Ashton-Tate, 9929 Jefferson Blvd., Culver City, CA 90230.

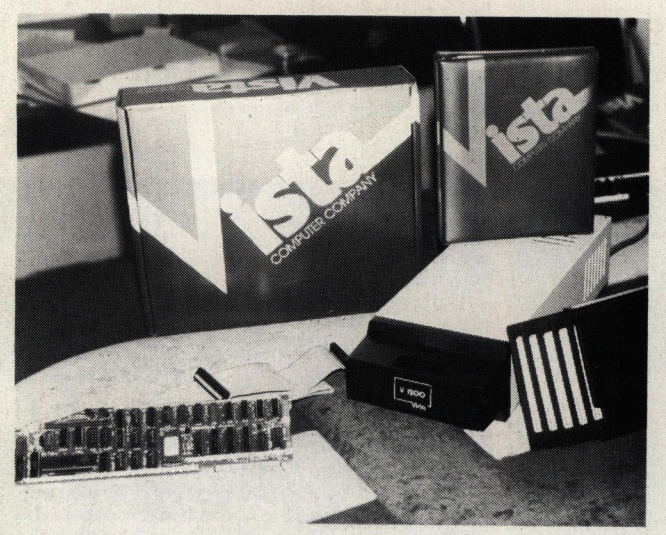
Ashton-Tate

©1982 Ashton-Tate
CP/M is a trademark of Digital Research

TOMORROW'S APPLES TODAY

V1200

Vista Computer Company Inc. introduces a new alternative to hard disk storage. The Vista V1200 offers the user, 6 Megabytes of removable, replaceable, interchangable storage in a plastic cartridge.



The V1200 uses the latest innovations in floppy storage technology to achieve 6Mb of storage in a pack of five 5¼" diskettes. The cartridge is slightly smaller than a standard box of ten minifloppy diskettes.

At $1549.00 (Sugg. Mfgr. list), the V1200 offers the user the speed and storage benefits of a hard disk, while retaining the removability and backup benefits of floppy.

For more information, please contact Vista Computer, 1317 Edinger, Santa Ana, CA 92705 (714) 953-0523, or an authorized Vista Dealer/Distributor.

AMPERWARE

Amperware is a software package which significantly extends the capabilities of Applesoft™ BASIC. Once **Amperware** is loaded, any of its many commands may be used in Applesoft™ programs. **Amperware** commands are much like Applesoft™ commands but are much more powerful. **&INPUT,** for example, allows upper/lower case character entry without additional hardware; accepts commas, colons, and quotation marks; limits the length of responses; controls the type (string, real, or integer) of response; supports default responses easily; intercepts illegal characters; provides full in-line editing (insert/delete characters, clear to end, restore default); allows special characters ([,], , , , ,|, , and _) to be entered and allows control codes to be processed easily. Another example of **Amperware's** powerful command set are its disk commands, **&READ** and **&WRITE.** With these commands you can: recall or store information up to *20 times faster* than with the standard DOS commands; handle commas in disk files; read or write entire arrays in a single statement; and save up to 75% on disk storage space. **&PRINT WITH** is **Amperware**'s answer to PRINT USING, allowing either text or numeric information to be easily formatted. Of course, **Amperware** allows GOTO and GOSUB to a variable, deleting of individual arrays, and fast sorting and searching of string or numeric arrays. **Amperware** also provides you with powerful commands to control the Apple 40 column or 80-column displays, allowing one program to work with either format.

An extensive tutorial and reference manual comes with **Amperware** to allow it to be used to its fullest potential. **Amperware** is provided on disk in standard DOS 2.2 format, and it is not copy protected. It may be used in any commercial program if a credit is given.

Amperware is available from dealers or directly from Scientific Software Products, 3171 Donald Avenue, Indianapolis, IN 46224 for $49.95 (add $1.50 shipping and handling).

PIEMAN

The penguins at Penguin Software were recently spotted (well, they weren't really spotted) migrating towards the vast Game Tundra — a place no penguin has ever been seen before. The strangest thing about these penguins, though, is that they were all wearing baker's hats! And as they waddled along they seemed to be changing something that sounded like MANPIEMANPIEMANPIE no wait, it was PIE MAN PIE MAN PIE MAN.

When we asked to explain this puzzling chant, the head penguin began to relate the tale of PIE MAN.

"You see," said the penguin, "PIE MAN works in a bakery. After the pies come out of the oven and onto a conveyor belt, it's PIE MAN's job to put whipped cream and a cherry on each pie then put it in the pie bin. PIE MAN dreams of becoming chief baker as he rushes around dodging flour sacks, grease spots and the slightly tipsy wedding cake baker. The pies come out of the oven faster and faster until. . .well, what ends up on the floor is not a pretty sight."

Then the penguin said, "We must continue our journey into the Game Tundra. PIE MAN is the first. There will be more to follow."

PIE MAN was written by Eagle Berns and Michael Kosaka for the Apple II+. It works with keyboard or joystick (including Atari joysitck) and retails for $29.95.

A STATISTICAL ANALYSIS SYSTEM FOR APPLE COMPUTERS

SoftCorp International, Inc. is introducing speedSTAT™, a major statistical analysis system for Apple II computers. Premiere showing will be at the "Applefest" computer show in San Francisco November 18-22. SpeedSTAT Volume 1: Frequencies & Crosstabs, has a capacity of over 10,000 data points and over 30 different statistical measures.

SpeedSTAT is very fast and very easy to learn, according to SoftCorp International Vice President Thomas E. Simpson. The content of the illustated user's manual, and the overall design of the software assume that the user has little or no computer experience. SpeedSTAT is designed for small business and professional users, including marketing professionals. It enables them to easily perform statistical analyses of demographics studies, product testing and market research data.

SpeedSTAT requires an Apple II or Apple II Plus with 48k and two disk drives. SpeedSTAT Volume 1 will retail at $250 and will be available in mid-November through Apple dealers nationally. SoftCorp International, Inc. has offices at 229 Huber Village Boulevard, Westerville, Ohio 43081 (phone 614/890-2820). For product information and orders, phone 800/543-1350.

CP/M UTILITY PROGRAMS

CP/M Disk Operating Programs are simple to use and need a minimum of learning to operate. It is the most adaptable system for many different computers you can buy on the market today. However, to work with CP/M programs some utility features could save time and effort. Programs include: Disk Zap, to examine and repair diskette information; Test Editor, to write and correct programs or letters and save on diskette as well; File Map, to show where program files are on the diskette; List, to show programs or file in ASCII form on video screen; Single Drive Transfer, copy file or programs with first drive only; Directory with Stat, show all files and programs their length and available space on diskette; Copy, copy diskette from one drive to another; Purge, to review one by one file on one diskette and erase those not needed; Tang DOC, documentation on all the programs.

The programs are now offered on one diskette for CP/M users. These programs are initialized with single key stroke for convenience. They will work in single, double and extended double density formats. To change computer program under CP/M, Tangdata also offers a diskette including unload command to change X.com to X.Hex then use Disasm.com to create source code ready for changes you want to make.

All utility programs are aggressively priced for mass distribution. Programs are available from stock: 8″ — $24.95 and 5.25″ — $19.95. Single Density Diskettes are suitable for IBMPC, TRS80's, APPLE, WANT, EAGLE, OSBORNE, HEALTH/ZENTH, etc. using CPM operating systems. For more information, call Tangdata Corporation, 100 Eames Street, Framingham, MA (617) 872-7520.

KALEIDO-SOUND

Passport Designs, Inc., announces their entry into the entertainment software business today with KALEIDO—SOUND™, a real time graphics program that synchronizes to any audio input.

The audio signal from a home stereo, cassette player or a sound system can be plugged into the Apple II cassette port to drive four different kaleidoscope patterns, each with a selectable color scheme. The full color kaleidoscopes change color, pattern, and location dynamically on a CRT monitor, television, or video screen as the music changes frequency and loudness.

KALEIDO—SOUND is targeted toward a variety of audiences, music listeners, computer hobbiests, game enthusiasts, night clubs, and concert acts.

KALEIDO—SOUND has a suggested retail price of $39.95 (including connecting cable) and is available at computer stores nationwide or direct from Passport Designs, Inc.

For more information please contact: Chris Albano, V.P. Marketing, Passport Designs, Inc., 116 No. Cabrillo Hwy, Half Moon Bay, CA 94019, (419) 726-0280.

KALEIDO—SOUND is a trademark of Passport Designs, Inc. Apple II is a trademark of Apple Computer, Inc.

SOSCLOCK /// CALIBRATION UTILITY

System Fabricators has released a new utility diskette for use with the SOSCLOCK /// onboard clock for Apple ///. The clock timebase provided on the Apple /// motherboard often requires calibration in order to obtain maximum accuracy from the SOSCLOCK ///, in the past this procedure required either expensive frequency counters or long trial-and-error methods.

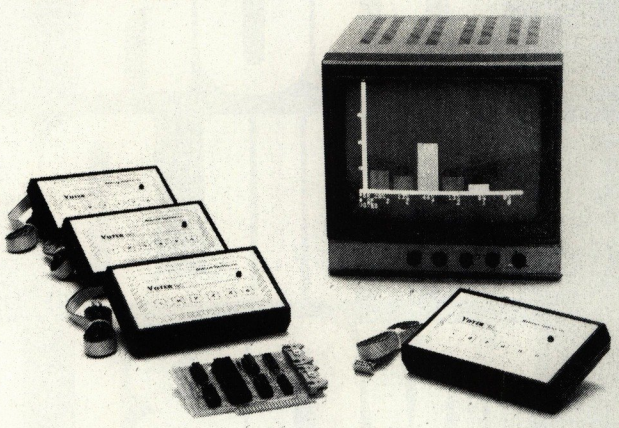
The calibration programs provided on the Utility diskette allow easy calibration of the clock speed using only a small screwdriver. Calibration to within a few seconds per month can typically be performed in less than ten minutes. Also included on the diskette are standard utilities for setting the clock from Apple /// Business Basic and Apple /// Pascal.

SOSCLOCK /// Calibration Utility V 2.2 is available as System Fabricators' part number 821021. Price is $20.00 per copy from Sysfab dealers or direct from: System Fabricators, 736 Hermosa Avenue, Hermosa Beach, CA 90254, 213-372-6273.

VOTER

VOTER 30 a new peripheral card links up to 30 participants to a single Apple II* computer.

VOTER 30 turns your Apple II* into a group response system, allowing up to thirty participants to respond to multiple choice questions—then letting them see the results immediately.



COMPLETE RECORDKEEPING — Each polling station, like each participant, is unique. VOTER 30 reports on group response and also keeps track of response by individual. This make VOTER 30 ideal for testing and market research applications.

SOFTWARE SUPPORT — VOTER 30 comes with BASIC language programs designed to help non-programming instructors write quizzes, retrieve and analyze test results. Diagnostic software will confirm whether the system is installed correctly.

FLEXIBILITY — VOTER 30 works in conjunction with other educational software, random access video systems, multiple video displays, etc. Polling stations are ordered in any quantity.

AFFORDABLE — The VOTER 30 interface card with all programs and manual costs $595. VOTER 30 polling stations cost $125 each and include cable and connectors.

Contact Reactive Systems, Inc., 40 North Van Brunt St., Englewood, NJ 07631, (201) 568-0481.

BYTE BOX

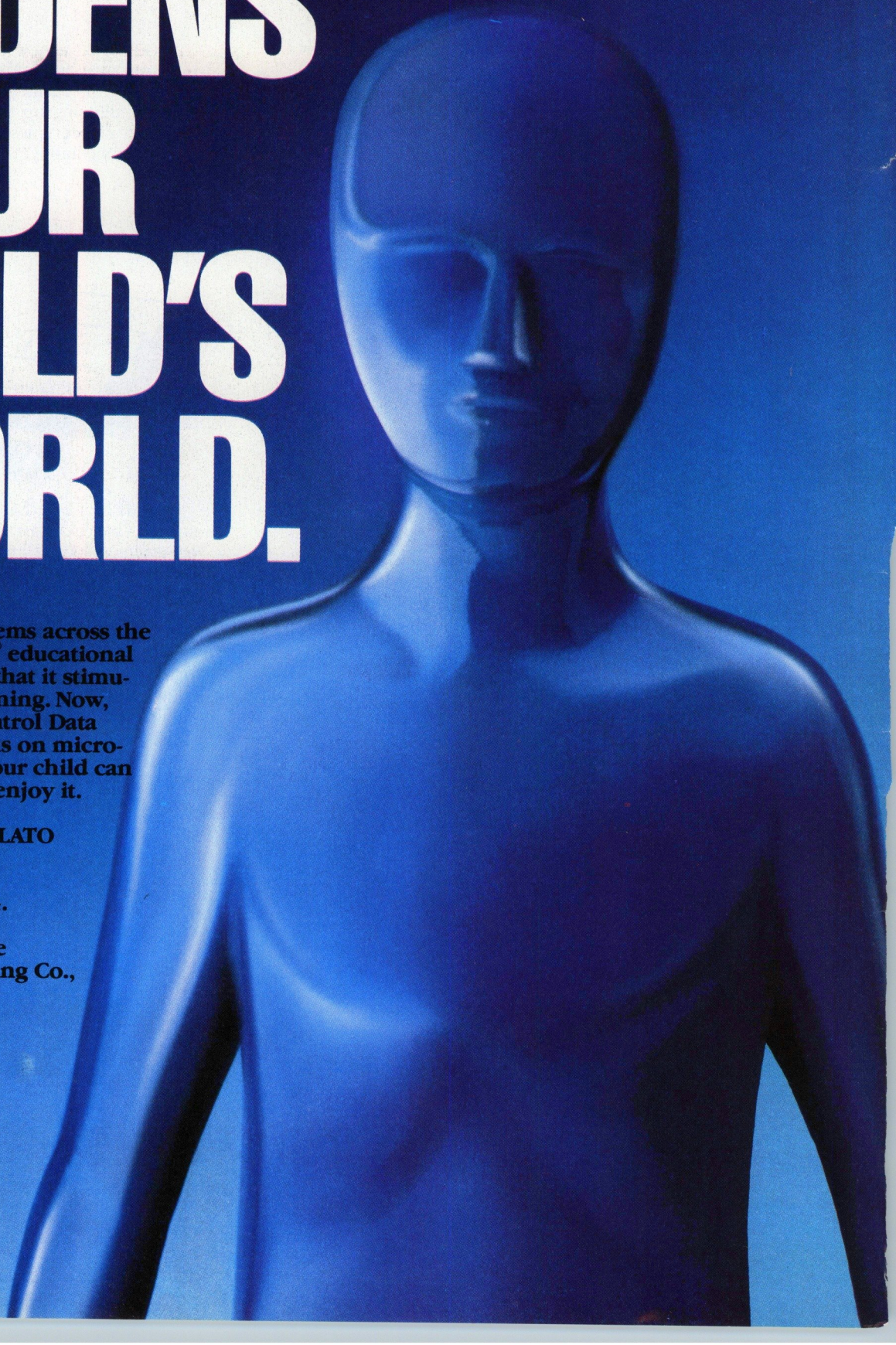
Designed specifically for your Apple II® and Apple II+®, the new Byte Box® offers you a variety of attractive features:

Program Protection — The Byte Box's® unique locking keyboard cover provides complete program protection. Only you have access to the keyboard, and if you have to leave your computer briefly while working, you can the lock the cover and leave your Apple II® and Apple II+® on.

Computer Protection — The Byte Box® has been designed to pamper your Apple II® and Apple II+®, with a powerful, built-in cooling fan and a special line-surge suppressor. In addition, there are plug-ins for your CRT and printer.

—◉→

PLATO WIDENS YOUR CHILD'S WORLD.



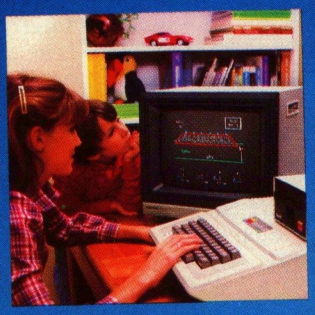
In public school systems across the United States, PLATO® educational software has proved that it stimulates and speeds learning. Now, for the first time, Control Data has put PLATO lessons on microcomputer disks, so your child can learn at home—and enjoy it.

For information on PLATO educational software, or to order, call toll free 800/233-3784. In California, call 800/233-3785, or write Control Data Publishing Co., P.O. Box 261127, San Diego, CA 92126.

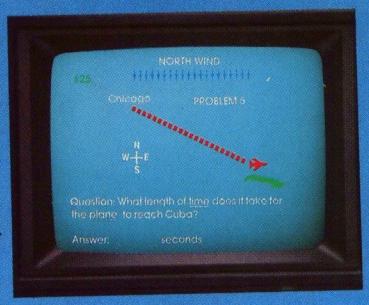
NOW, QUALITY EDUCATIONAL SOFTWARE IS AVAILABLE FOR YOUR HOME COMPUTER.

- PLATO® software now available on disks in Math, Physics, Foreign Languages, Computer Literacy—and more is on the way.
- For use on your Apple II Plus, TI 99/4A or Atari 800.
- Introductory Offering: Single lesson, $45.00; additional lessons, $35.00 each. 10-day money-back trial.

CHOOSE THE LESSONS THAT FIT YOUR CHILD'S NEEDS.



- **Basic Number Facts:** lets your child practice basic numbers, including addition without carrying, subtraction without borrowing and multiplication/division with single digits. Kids race against time to build up their speed in these math areas. Designed for elementary school age, or any child who needs practice with basic whole number operations.
- **Whole Numbers:** simulates a pinball game to hold and build interest in whole number operations. Problems include addition, subtraction, multiplication, division and mixed numbers. Designed for elementary and junior high age groups.
- **Decimals:** kids are challenged to break balloons by entering the correct decimal corresponding to the position of the balloon on a vertical line. Numbers are entered on a trial-and-error basis. Software automatically adjusts difficulty to the child's performance. For elementary math students.
- **Fractions:** same format as Decimals, but requires the use of fractions to break the balloons. Numbers are entered on a trial-and-error basis. Balloons may burst in any order until none are left. Difficulty adjusts to your child's performance. For elementary math students.
- **Computer Literacy—Introduction:** this lesson is presented in a friendly, non-intimidating manner with touches of humor and simple, supportive graphics. It presents the history and uses of computers in today's society. Designed for junior or senior high and vocational school students.
- **French Vocabulary Builder.**
- **Spanish Vocabulary Builder.**
- **German Vocabulary Builder:** students are presented with a basic vocabulary of 500 words, including useful verbs, number words or common words for traveling or in the home. Groups of related words give learners context and similarity clues, which help increase foreign language skills. Lessons supplement introductory and/or refresher coursework.
- **Physics—Elementary Mechanics:** provides a problem-solving test in the elementary mechanics of physics. Students are shown a physical problem; then must "purchase" the missing information needed to answer it correctly. The emphasis is on *understanding* the problem, rather than just supplying the correct answers. Designed for senior high age level.



CONTROL DATA

PLATO

COMPUTER-BASED EDUCATION

TOMORROW'S APPLES TODAY

Good Looks — The new Byte Box® is available in a wide variety of beautiful, hand-rubbed natural woods and a complete range of colorful simulated-leather laminates. And the sliding cover lets you transform your Apple II® and Apple II+® into a beautiful-crafted room furnishing in just seconds.

Beauty, Protection and Price . . . It's an open and shut case for your Apple II® and Apple II+®! The price is $289.00 with line suppressor, fan and plug-in. $199.00 without the above parts. Add $15 for Walnut finish and $20 for Teak.

From KEL TECH, 34732 Calle Fortuna, Capistrano Beach, CA 92624, (714) 661-0435.

SUPER SMART

SUPER SMART — smart terminal software for the APPLE II. SUPER SMART is a sophisticated terminal emulation package that utilizes the full capabilities of the D.C. Hayes Micromodem II for terminal to terminal and terminal to time-share computer communications. SUPER SMART can capture, create, edit, print, and save to disk up to 21,224 bytes of data. SUPER SMART can send or receive telegrams, HiRes pictures, BASIC or machine language programs, as well as canned messages and editor files.

SUPER SMART supports a full ASCII (upper and lower case) HiRes character set that can be customized, and a software shift-lock for ASCII or standard APPLE keyboard emulation.

A unique feature of SUPER SMART is its ability to access and maintain any number of phone list files each containing more than 500 entries. Any combination of up to 10 characters can be specified as a search key for which SUPER SMART will look-up and present all phone list entries containing the search key and will automatically dial selected entries. Manual phone number specifications with redial capability is provided for numbers not contained in the phone list. An instant access default phone number can be programmed-in to save time and keystrokes.

Requires: HAYES Micromodem II, 48K APPLE II/III+, 1 DISK II. Price: $60 from Softspoken, P.O. Box 7000-863, Redondo Beach, CA 90277.

SBCS AGRI-LEGER

SBCS Agri-Ledger is a double entry accounting system for agricultural operations, designed for the Apple II and III. This menu-driven program provides for income statements and balance sheets, budget reports, comparison of performance against projections, enterprise analysis and report combinations for up to 26 different operations with up to 99 different enterprises on each, quantity fields, Schedule F information, yearly history, and much more. Cash flow templates are included for both farm and personal items. The documentation, contained in an attractive self-standing binder, is detailed and easy to understand. Program backup disk is included. Demo package is available and comes with full documentation; price of the Demo can be credited towards purchase of full program. SBCS provides a lifetime warranty, free program updates and replacement of damaged disks, and friendly customer service. Additional ag modules which integrate with Agri-Ledger will be released at a later date. Agri-Ledger —$395. Demo — $30. Small Business Computer Systems, 4140 Greenwood, Lincoln, NE 68504 (402) 467-1878.

FILE-FAX™

NEW IMPROVED **DATA BASE MANAGEMENT SYSTEM**

The easiest-to-learn, simplest-to-use DBMS available today! Designed as a "filing system" for the businessman or hobbyist, FILE-FAX quickly accesses records, retrieving information at exceptionally high speeds.

FILE-FAX can be used to keep track of a wealth of information—about people, places, dates, events. It has an 8-level sort, wide ranging search capability, and a powerful report generator. Use it for inventory control, customer files, mailing lists, purchase records, and more—you are limited only by your own imagination.

FILE-FAX will run on your Apple II or II+, Atari 800, IBM-PC, NEC PC-8001, Commodore 64, Osborne, and Victor 9000. Write for full details. Please specify the computer you are working with.

Trade Mark of Quality **tmq SOFTWARE, INC.**

Dept. C, 82 Fox Hill Drive, Buffalo Grove, IL 60090

A.P.P.L.E. Members Only

You've Seen the Rest Now Get the Best

GLOBAL Program Line Editor

Global PLE has all these fine new features to further ease your editing tasks:

- Global edit, search and replacement
- Compatible with 80 column cards
- Typeahead buffer
- 1152 bytes of Escape functions definable from keyboard and nestable
- Page list mode
- Completely transparent to user
- Loads in regular RAM or Language card.

Plus all of the powerful standard PLE commands and features

$38.50

A.P.P.L.E. Orders
304 Main Ave. S., Suite 300
Renton, WA 98055
(206) 271-4514

Washington state residents add 7.0% sales tax
Foreign members add $12.00 shipping and handling charge
Canadian members, postage being revised

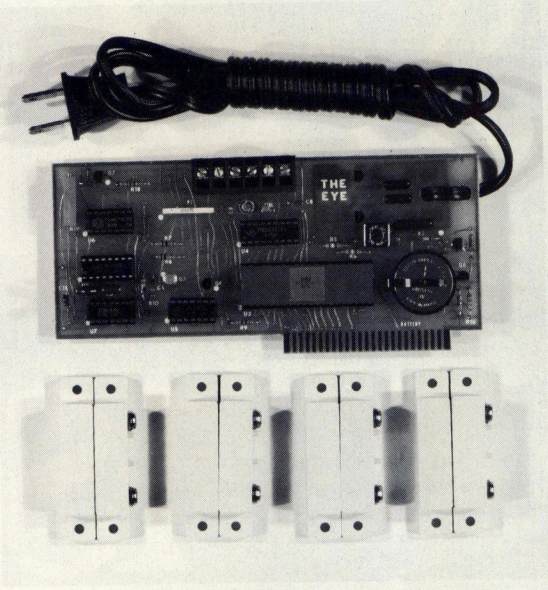
PROTECTION

FOR YOUR HOME & BUSINESS

Turn your Apple Computer into a sophisticated burglar detection system with THE EYE... the only accessory which provides you and your family with complete, round-the-clock security at a price far below that of conventional alarm systems...

ONLY
$199
COMPLETE

- On-board REAL TIME CLOCK activates the EYE at user-specified time intervals.
- On-board BSR REMOTE CONTROLLER automatically turns house lights on and sounds alarm upon illegal entry.
- Fully programmable LIGHT TIMER feature gives your residence a "someone is home" appearance while you're away, discouraging any burglary attempt.
- PHONE DIALING CAPABILITY—relay circuit triggers optional dialer to notify police or store owner of a break-in.
- 3 INDIVIDUAL ALARM CIRCUITS allow each floor or area to be monitored separately under independent time control.
- Complete line of inexpensive DETECTION SWITCHES available for every application—including fire detectors!
- Complete Software Package included.
- REAL TIME CLOCK and REMOTE CONTROLLER can be used separately, independent of the alarm—it's like three devices in one!
- EASY user installation—no professional contractor needed!



THE EYE COMES COMPLETE WITH 4 MAGNETIC ENTRY SWITCHES, USER MANUAL, AND SOFTWARE ON DISK. ALL FOR ONLY $199. DON'T LEAVE YOUR HOME OR BUSINESS UNGUARDED...

ORDER YOURS TODAY!!

See your authorized dealer or contact us:

LEHIGH VALLEY COMPUTER CORP.
523 So. Clewell St., Bethlehem, PA 18015

_____ I want to protect my home or business. Send me **THE EYE** for only $199.

_____ I'd like more information about **THE EYE**

Name _____

Address _____

City, State, Zip _____

☐ Check ☐ M.O. ☐ C.O.D. ☐ VISA ☐ MC

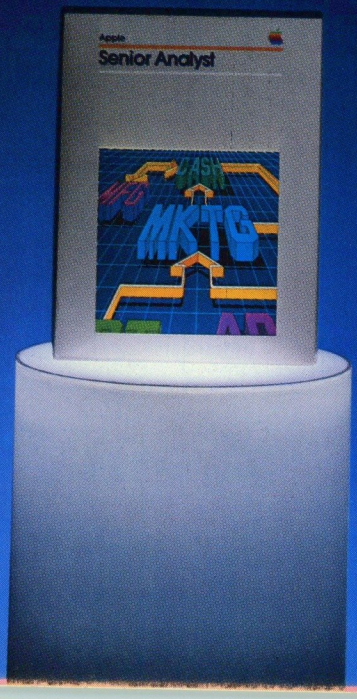
Card No. _____ Exp. Date _____

CALL TOLL FREE
1-800-325-9800
IN PA. CALL COLLECT
215-868-1303
VISA AND **MASTERCARD** ACCEPTED
Dealer Inquires Invited

COMING SOON — **THE EYE** FOR THE **ATARI** AND **COMMODORE VIC-20**

REMEMBER...THE EYE NEVER SLEEPS!

What if "what if" isn't enough?



Just when the business world is up to its white collars in visiclones, calcalikes and other spreadsheet packages, Apple's® come out with something entirely different.

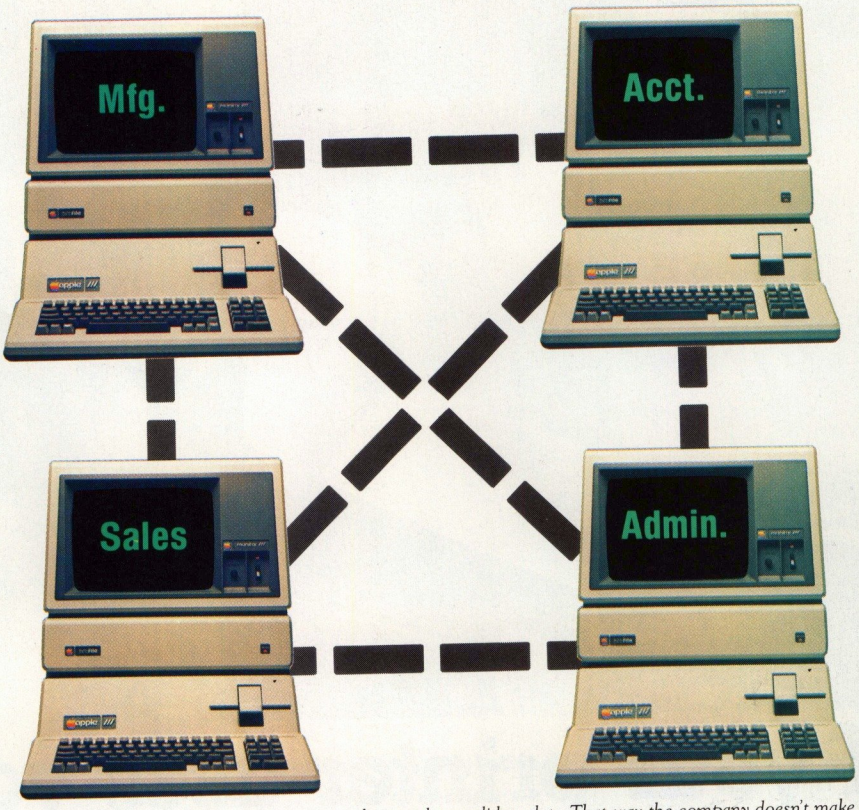
Introducing Senior Analyst.

Like other financial modeling packages, it allows managers and professionals to ask all those proverbial "What If?" questions.

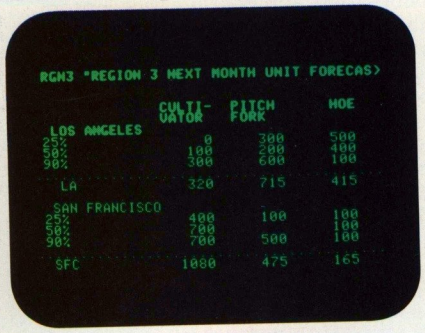
Unlike the others, this powerful financial planning tool was designed to be used in a corporate environment, by lots of people. So you get lots of advantages.

For example, you can transfer data (across diskettes) from one financial model to another. Or consolidate many models into one.

So sales, manufacturing, administration and any number of other departments (even in other cities) can easily share information. Giving each the power to create comprehensive and



Senior Analyst lets different departments share and consolidate data. That way the company doesn't make more pitchforks than it can sell.

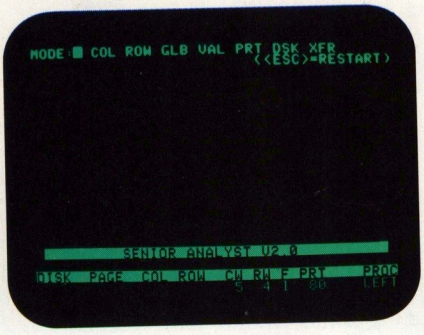


With Senior Analyst, you can now cultivate forecasts by merging reports from distant divisions.

flexible financial projections, budgets, cash flow statements and the like.

Want to combine selected data (such as important subtotals) from six different divisions? With Senior Analyst, you can do it. And even print out a formatted report that includes only the information you need.

A report that anyone can understand. Because the headings are in English, not in code.



Easy to follow commands allow employees to create models without learning a second language.

You can also document and print out all those assumptions used to create your model, to give others a concrete understanding of how you reached your conclusion. (The program even allows you to continue working while a model is being printed.)

To complement all these accommodating features, you'll also find built-in functions for depreciation, linear regression forecasting, and other powerful virtues not found in most financial software packages.

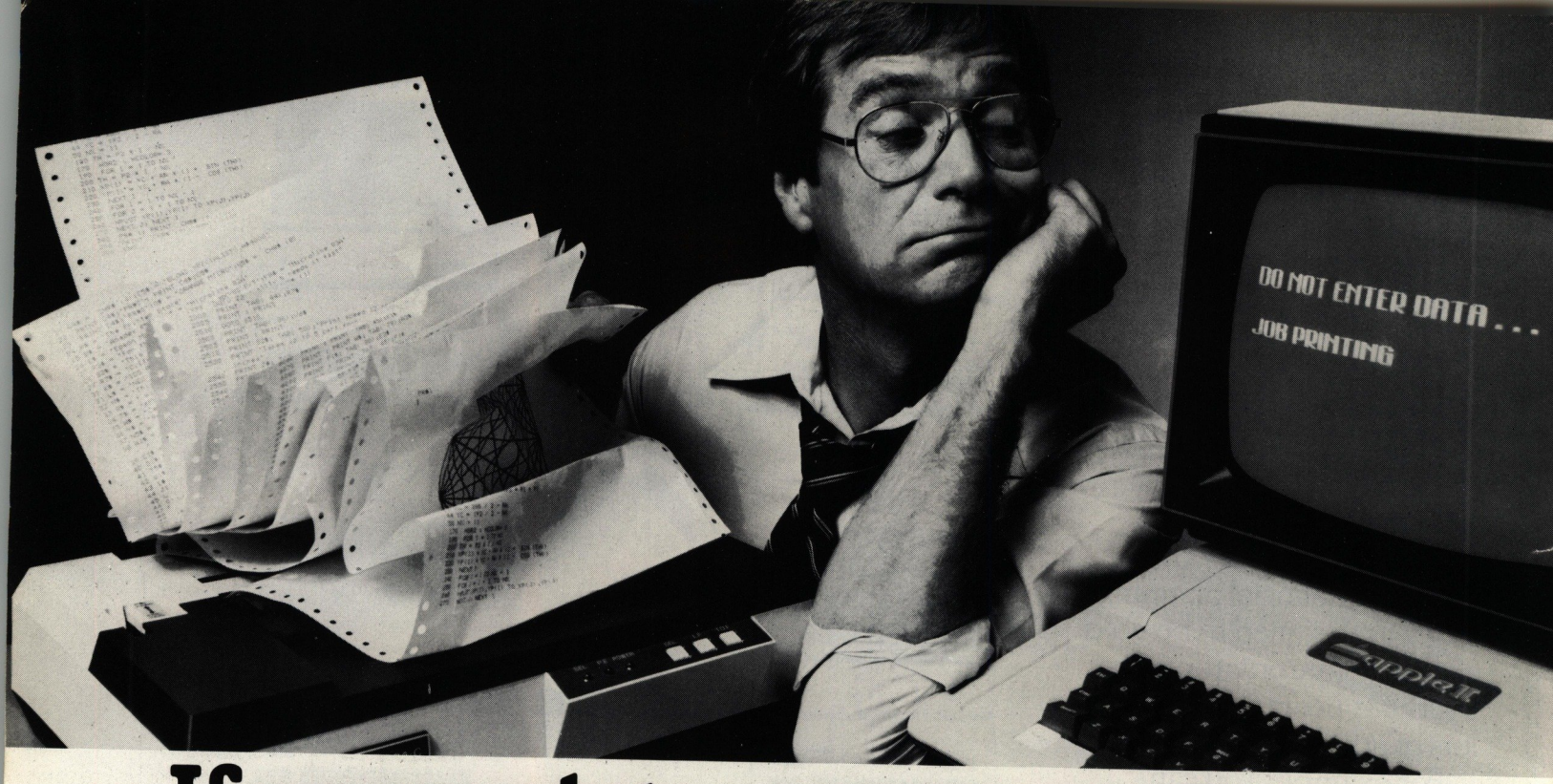
All of which we'd like you to experience in person, at any of our 1300 authorized full-support dealers (they also offer a vast library of other quality software distributed by Apple for Apples).

And don't ask for just any spreadsheet package. Tell them you need to see an analyst.

apple®

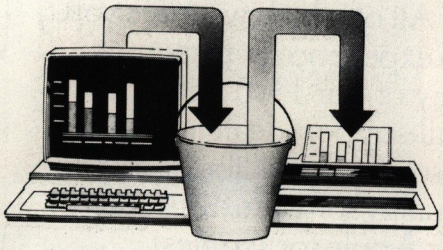
The most personal software.

Call (800) 538-9696 for the location of the authorized Apple dealer nearest you, or for information regarding corporate purchases through our National Account Program. In California (800) 662-9238. Or write Apple Computer Inc., Advertising and Promotion Dept., 20525 Mariani Ave., Cupertino, CA 95014. © 1982 Apple Computer Inc.



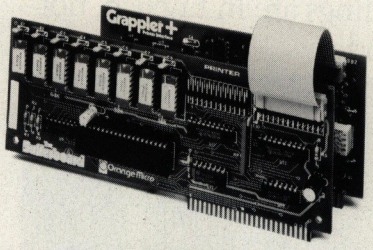
If your printer uses your Apple® more than you do, you need The Bufferboard.™

If your Apple is locked into the "PRINT" mode so much that you've taken up solitaire to kill the boredom, you need a buffer. And if your computer is the Apple II or III, the only buffer for you is The Bufferboard. Expandable to 64K of storage, The Bufferboard stores an instantaneous **bucketful** of print data from your computer. Then it feeds the data to your printer at its own printing rate. Your Apple is set free from driving your printer and is ready for more data from you.



Take your existing interface—and buffer it!

Only The Bufferboard has a simple Interface-Docking System. No bulky boxes or expensive power supplies are needed because The Bufferboard fits right into your Apple—and **docks** onto your existing printer interface. The result is convenient



and economical buffering of most popular printer interfaces, including the Grappler + ™ interface, Epson interface, and Apple printer interface. Thirty seconds and a single hook-up are all you need to end the printer waiting game forever.

Up to 20 letter-size pages stored at a time.

The Bufferboard comes standard with 16K, and is expandable to 32K or 64K of buffering capacity with the addition of memory chips. This "bucket" will hold up to 20 pages of a print job, allowing you freedom to use your Apple.

The Bufferboard—designed exclusively for the Apple Computer.

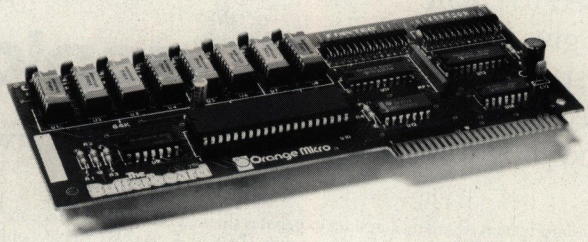
Specifications:

• Versions for Grappler + interface, Epson interface, Apple interface, and other popular printer interfaces • 16K buffer standard • Upgradeable to 32K or 64K • Automatic memory configuration • Automatic self test • Includes interface docking cable.

The Bufferboard is made by Orange Micro, Inc.; the same people who brought you the popular Grappler + printer interface. Both the Grappler + and The Bufferboard are now available at your local Apple dealer.

Apple is a registered trademark of Apple, Inc.
Epson is a registered trademark of Epson America, Inc.

Orange Micro inc.
3150 E. La Palma #G, Anaheim, CA 92806
(714) 630-3620, TELEX: TX 183511 CSMA



The Bufferboard™
For Apples and Printers

© Orange Micro, Inc. 1982

CONSULTANTS

Bob Huelsdonk	Apple ///(Western U.S.)	(206) 362-4910
Stuart Yusem	Apple /// (Eastern U.S.)	(609) 822-9306
Erv Edge	Assembly (Alaska)	(907)337-7620
Dennis Baumgarte	Assembly (Eastern U.S.)	(914) 473-3210
Don Williams	Assembly (Western U.S.)	(206) 242-6807
Michael Thyng	Business	(206) 524-2744
Steven E. Bair	Data Bases	(913) 782-5901
William P. Steinberg	DOS (Eastern U.S.)	(212) 921-0663
Art Schumer	DOS (Northern U.S.)	(701) 282-7907
Lee Meador	DOS (Southern U.S.)	(817) 469-6019
Kurt Thams	DOS/Gen'l (Western U.S.)	(714) 870-9316
WM. Reynolds III	DOS/Tech (Western U.S.)	(503) 472-4028
Ted Perry	Education	(916) 485-1690
Murray Arnow	Engineering Applications	(312) 291-9210
Cornelis Bongers	European Consultant	010-525511 Ext. 3275
Alan Robinson	Fortran	(503) 757-8234
Jim Eatherly	Games	(202) 232-6046
Stan Axelrod	Hardware Design/I-O Control	(415) 488-0915
Tom Deal, M.D.	Handicapped Applications	(206) 634-2847
Randy Halvorson	Hardware Interfacing	(206) 783-9040
Ed So	Hi-Res	(408) 267-3448
Steve Ross	Mainframe Interfacing	(206) 838-2810
Stowe Keller	Mathematics	(804) 973-5701
Fred Merchant, M.D.	Medical Group	(415) 921-7792
Stephen F. Lee	Music	(704) 289-4102
Tom Woteki	Pascal (Eastern U.S.)	(202) 547-0984
Carl Winter	Pascal (Western U.S.)	(206) 454-2720
Randy Warrens	Pilot Language	(608) 837-9566
Hersch Pilloff	Stock Market	(301) 292-3100
Mark Robbins	Telecommunications	(303) 755-6440
Clifton Howard, M.D.	Utilities	(201) 768-9274
Jim Hockenhull	Video Interfacing	(509) 334-4137
Jerry Brieger	VisiCalc (West)	(206) 881-1241
Vince Aderente	VisiCalc (North)	(312) 289-9685
Dr. Bob Richard	VisiCalc (South)	(504) 466-0894
Brian Abrahams	VisiCalc (South Africa)	(Durban) 315442
Jeff Finn	Word Processing	(206) 325-0430

AD INDEX

A.P.P.L.E. 39, 40, 52, 53, 90
Action Research NW 35
Adventure International 44
Apple Computer Inc. 92, 93
Applied Analytics 41
Ashton-Tate 84, 85
Assembler Teacher 20
Balbes Software System 43
Beagle Brothers 60, 61
Bit 3 Computer Products 76
Bytes & Pieces 32
Cases, Inc. 55
Central Point Software 62
Computer Case Co. 74
Computer Consulting Services 69
Connecticut Information Systems 24
Control Data Corp. 88, 89
Cortland Data Systems 39
Datamost 1, 81
Dickens Data Systems 52, 78
Educational Computing Systems 54
Frobco 57
Futra Company 30
Gourmet Software 35
High Order Micro Electronics 29
Ivers Specialties 68
Leading Edge Products ..... IFC, BC
Lehigh Valley Computer Corp. 91
Micro-Ware Distributing Inc. 54
MPC Peripherals 42
Northern Technology Books 66
Northeast Expositions 38
The Olive Branch 40, 66
Omega Microware Inc. 12, 40, 48
Orange Micro 56, 94
Passport Design ..... IBC
Personal Computing Products 6
Pirates Harbor 47
Proline Electronics 78
R. C. Electronics Inc. 80
R. H. Electronics 8, 69
S & H Software 59
Sleeping Bear Software 73
Softkey Publishing 74
The Software Farm 13
Southeastern Software 14
Southwestern Data Systems 22, 36
SRA 51
SSM 62, 73
Stellation Two 46, 96
Star Logic 74
Sympathetic Software 66
Telephone Software Connection 69
Texas Ranch & Shoreline System 64
TMQ Software 90
Vista Computer Co. 4
XPS, Inc. 36

A.P.P.L.E. ORDER FORM

DATE _____

MEMBERSHIP NUMBER _____ EXPIRATION DATE _____

NAME _____ ADDRESS _____

CITY _____ STATE _____ ZIP _____ PHONE _____

PRODUCT	UNIT PRICE	*FOREIGN SHIPPING CHARGES (See below)	AMOUNT

VISA* Account No. _____ Subtotal _____

Exp. Date _____ **Washington sales tax _____

MasterCard Account No. _____ TOTAL AMOUNT _____

Exp. Date _____

FOREIGN SHIPPING/HANDLING CHARGES CHART

In an attempt to simplify things for our members outside the United States, we have devised this easy rate chart for figuring the shipping and handling charges on your order. Simply determine the **TOTAL** amount of your order and add the correct amount as corresponds to the rate chart. (NOTE: THIS DOES NOT APPLY TO APPLICATIONS OR RENEWALS — BUT TO PRODUCT ORDERS ONLY!)

total amount of order	shipping/handling
$ 0.00 - $ 20.00	add $ 6.00 to order
$ 20.01 - $ 60.00	add $12.00 to order
$ 60.01 - $100.00	add $18.00 to order
$100.01 - $140.00	add $24.00 to order
$140.01 - $175.00	add $30.00 to order
$175.01 - and over	add $36.00 to order

Note to Foreign Members: All hardware products, except AP.LCASE lower case adapters, must include a United States address. These products cannot be shipped outside U.S.

* Foreign Members Only

** Washington state residents add 6.5% state sales tax.

SEND TO: A.P.P.L.E.
304 Main Ave. S.
Renton, WA 98055

***NOTE: Canadian shipping/handling chart is being revised.**

REQUEST FOR CHANGE OF ADDRESS

Dear A.P.P.L.E.,

I am moving and my present address is:

PRESENT:

NAME _____

ADDRESS _____

CITY _____

STATE _____ ZIP CODE _____

EFFECTIVE DATE _____

NEW ADDRESS:

NAME _____

ADDRESS _____

CITY _____

STATE _____ ZIP CODE _____

MEMBERSHIP NUMBER _____

*Please fill out and return this notice two weeks before you move.

UNCLASSIFIEDS — A.P.P.L.E. ORDERS

CALL -A.P.P.L.E. (un)Classifieds

Members **Non-commerical** rate .05/word
Members commerical rate .25/word
All others: 1.00/word, $25 minimum
Send ad copy with payment to editorial office

APPLE EDUCATION: PHYSICS — 11 disks, 75 programs, extensive graphics. Used in over 600 schools. $200/set. Vectors $10; Statics $12; Motion $12; Conservation Laws $12; Circular Motion $15; Thermodynamics $20; Electricity $12; Optics $20; Atomic Physics $30; Solar System $30; Stellar Astronomy $30. Other programs: AQUARIUM $25; DINOSAURS $15; BLITZKRIEG $15; JONAH $6; BIBLE BOOKS $15; THE CHRISTMAS STORY $12; GRADE REPORTER $19.95; PEACHY WRITER (an easy to use text editor) $24.95. VISA/MC. Free catalog. Write Cross Educational Software, 1802 N. Trenton, Box 1536, Ruston, LA 71270. (318) 255-8921.

FOR APPLE PLOT, YOU NEED INTERFACE. Interface translates from Visicalc to Apple Plot, prevents erroneous graphs, fits curves to data, and supplements Visicalc with rank ordering and alphabetizing. Send SASE for details or send $30 for the copyable program. Bill Starbuck, 2100 E. Edgewood, Shorewood, WI 53211, (414) 963-9750.

VEECEE-WRITER TRANSLATES VISICALC (/PF) FILES FOR APPLE WRITER 1. Send $15 for the copyable program. Bill Starbuck, 2100 E. Edgewood, Shorewood, WI 53211, (414) 963-9750.

"SAVE TIME, MONEY, GAS! Use ROUTEPLANNER with your maps to quickly find the order of stops and distance for the shortest route! Excellent for your multistop business, sales, pickup, delivery, bus trips, etc...For FREE information write: ROUTEPLANNER, Columbia Software, Box 2235E, Columbia, Maryland 21045. (301) 997-3100"

(6) 16 K RAM CARDS - NEW — $49. each 1 year guarantee. (1) Epson MX Printer — used one week — with interface card and cable $425. (206) 226-6999.

APPLE PASCAL ANIMATION GRAPHICS. Morwe/Graphics - 83 pp. manual and two diskettes. SASE for details, $3.95 for manual, **-$19.95-** for complete kit. HENDERSON ASSOCIATES, 980 Henderson Ave., Sunnyvale, CA 94086.

SEE THE CORE OF YOUR APPLE! Just released: A 400 square inch poster of the 6502 CPU's micro topology. See the Apple's microprocessor enlarged over 20,000 times in full color. You've played with it, programmed it, and cursed at it. Now discover what the little beastie looks like. Search for that infamous "bit bucket" or try to find out where all those software bugs hide. Each order includes a free fact sheet that explains chip history, component location and topology. satisfaction guaranteed, will arrive undamaged. Poster only $4.50. Omicron Digital Systems Ltd., P.O. Box 3055, Falls Church, VA 22043.

MUST SELL APPLE II EQUIPMENT WORKING: 48K Motherboard $450. Integer or Applesoft ROM Board $110. Communications Board $110. Disk II Analog Board $110. Add on SA400 drive for Apple $325. SA400 analog board converted for Apple use $65. 16K RAM Board $95. Replacement Keyboard $150. Drive case $39. Integer ROM set with aid #1 $35. Applesoft ROM set with autostart $45. MC multifunction card with software/manual $140. Eurapple II motherboard, no TTL, RAM, ROM $225. Much, Much, More! NON-WORKING: Mountain Computer (MC) Music System (boards only) $95. MC A/D-D/A converter $95. MC bare expansion board $95. NON-WORKING NO-TTL: Apple II motherboard $250. Serial, Parallel, Communication, Language, ROM, or Disk analog board $45. ANY THREE $120. ALL SIX $200. NO REASONABLE OFFER REFUSED. RICHARD SMITH, 2961 S. Las Palmas, Mesa, AZ 85202 (602) 820-0082.

I AM INTERESTED IN CONTACT WITH APPLE USERS involved with LISP. For about a year I have been working with Datasoft's Interpreter of LISP and have some material to share with "anybody out there." Ashville School, Ashville, NC 28806 (704) 254-6345.

FOR SALE: Apple Disk II w/o Controller $200. Call 415-856-8147.

VOTRAX TYPE'N TALK voice synthesizer. State of the art in voice synthesis without special programming. $375 new, will sell for $200. 206-752-2009 eves.

APPLE SOFTWARE RENTAL & sales. Over 2500 programs available. Rent that expensive software for just pennies a day! Digital Delli Software. 2961 S. Las Palmas, Mesa, AZ 85202 (602) 820-0082.

AMATEUR RADIO OPERATORS!— The Radcom Plus+ Package consists of a quality TU interface that installs in the Apple and connects to your rig. Included is SUPER-RATT, the most advanced, feature packed software ever developed for Sending and Receiving RTTY and Morse Code. Features like Selcall, Radio Bulletin Board, Verified File Transfer and much more! Detailed information from: Alex M. Massimo — AF6W, 4041 - 41st Street, San Diego, CA 92105.

ANNOUNCING ROADSEARCH for your trips. This powerful program computes the shortest practical auto routes between cities in the USA/Canada. . .and more! Database contains 406 cities and 69,000 road miles. Outputs include route, mileage, time and gallons. Printer optional. Ten day moneyback GUARANTEE. Only $34.95 (plus $1.50 s/h). Check/VISA/MC. Columbia Software, Box 2235K, Columbia, MD 21045. (301) 997-3100.

APPLE TEXT FILE TUTORIAL: Step-by-step disk-based beginner's introduction to Apple sequential text files. Teaches you to read, write, use & EXEC. DOS 3.3, 48K. $15.00 (Calif. add 6%). Send check to Computer Explanations, 2438 La Condesa Dr., Los Angeles, CA 90049.

A 16 Bit "Mind" and An 8 Bit "Body"

That's what you get with **THE MILL.** The most effective combination of 16 and 8 bit characteristics for the Apple II/II+. For those who mean business!

STELLATION TWO

The Lobero Building P.O. Box 2342
Santa Barbara, Ca. 93120
(805) 966-1140 Telex 658439

PRODUCTS NOW AVAILABLE TO A.P.P.L.E. MEMBERS

Call —A.P.P.L.E. Specials

Word Weaver Special[1]	$60.00

A.P.P.L.E. LIBRARY

A.P. Drive Technical Manual	$30.00
"Apple Orchard" No. 1 (magazine)	$2.50
Apple Users Guide	$13.00
Assembly Lines: The Book	$16.50
Beneath the Apple DOS	$17.00
"Best of Cider Press"	
(S.F. Newsletter Annual) Vol I	$4.50
(S.F. Newsletter Annual) Vol II	$9.50
Call —A.P.P.L.E. annuals	
1978	$7.50
1979	$10.00
1980	$15.00
1981	$18.50
Call —A.P.P.L.E. In Depth No. 1	$7.00
Call —A.P.P.L.E. In Depth No. 2	$11.00
The Elementary Apple	$11.00
Inside Washington Apple Pi	$7.00
Kids and the Apple	$16.00
Power of VisiCalc Vol I or II, each	$8.50
Using 6502 Assembly Language	$17.00
What's Where in the Apple	$12.50
Woz Talks (VHS Video cassette)	$27.50

AS IS SOFTWARE

Public domain diskettes (DOS 3.2.1) (Send 37¢ S.A.S.E. for list of titles)	ea. $4.00

A.P.P.L.E. DISKPAKS

Call —A.P.P.L.E. Anthology Vols I-VII, incl.	ea. $12.50
DICE 82[7]	$12.50
Diskpak 12	$16.50
Diskpak 34A	$13.00
Diskpak 5	$13.00
Diskpak 6B (DOS 3.2 disk utility)	$18.50
Diskpak 6B-16 (DOS 3.3 disk utility)	$18.50
Diskpak 7	$17.50
Diskpak 8	$13.00
In Depth 1 Diskette	$16.50
In Depth 2 Diskette Set	$20.00
Pascal Anthology '81	$12.50

A.P.P.L.E. HARDWARE

ApController	$95.00
Ap.Coolkit	$22.50
AP.DRIVE	$325.00
AP.DRIVE with Controller	$420.00
Ap L Case 1	$30.00
Ap L Case 2 (Lower case chip)	$27.50
Disk Banks	2 for $9.00

AP•Jewelry

Lapel pin	$3.00

Apple Prestige Product LinE

APfile — Text file utility	$18.50
Applesoft Carpenter	$18.50
Aptest	$18.50
Big Mac Macro-Assembler/Ted	$22.50
Big Mac.LC	$28.50
Big Mac X-Ref	$7.50
Global Program Line Editor	$38.50
Higher Fonts 1	$13.00
Higher Text Plus	$27.50
Program Global Editor	$18.50
Program Line Editor	$20.00
Soft Seventy	$22.00
THE Spreadsheet[2]	$22.50
Symbol Symon	$20.00
WordWeaver ///	$75.00

[1]Expires January 31, 1983
[2]Requires 16K RAMCARD. Also: must be ordered in separate envelope (no other orders included).
[7]$7.50 With Ap L Case 1
$10.00 With Ap L Case 2

Diskettes supplied on DOS 3.3 except as noted.
All walkin orders must be called in prior to pickup.

NOTICE TO FOREIGN MEMBERS:
A.P.P.L.E. Hardware products not available to overseas or foreign members. This includes Ap.Drive & Ap Cool Kit.

ORDER FORM ON REVERSE

entertainment fit for a muse.

KALEIDO-SOUND. A new concept in entertainment software from Passport Designs. Simply plug your stereo, cassette player or sound system into your Apple II™ and let KALEIDO-SOUND turn your music into brilliant full color graphics.

Imagine – four different kaleidoscopic patterns with selectable color schemes pulsating in perfect harmony with your favorite music. Add a fantastic new dimension to parties, concerts or your home entertainment system.

KALEIDO-SOUND runs on a 48K Apple II plus. Plug an audio signal into the cassette port and see what you've been missing.

KALEIDO-SOUND. Made for mortals but fit for a muse.

$39.95

Add $1.00 for direct orders.
VISA, MASTERCARD, check or COD accepted.

KALEIDO-SOUND is a trademark of Passport Designs, Inc.
Apple II and Apple II plus are trademarks of Apple Computer, Inc.

Get it at your local dealer or order direct from:

PASSPORT DESIGNS INC

116 North Cabrillo Hwy., Half Moon Bay, CA 94019

(415) 726-0280

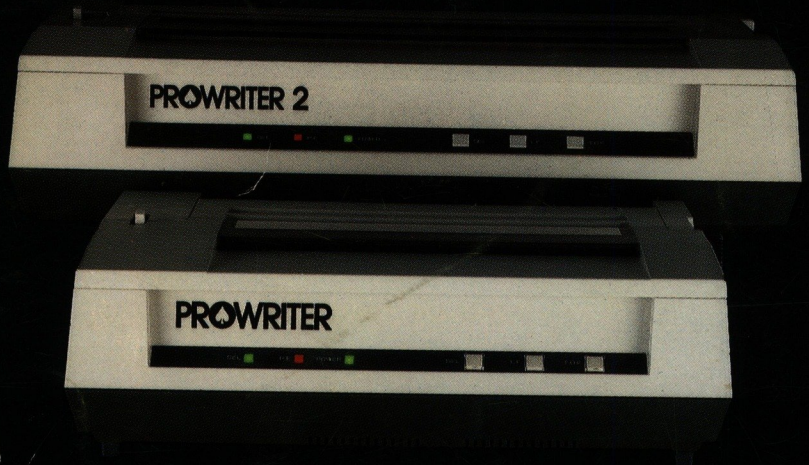
THE LEADING EDGE IN PRINTERS

ONE GREAT LINE. ONE GREAT WARRANTY.

Finally, there's one full family of printers that covers every business or word processing application—all from C. Itoh, a company known for packing more product into less price; and all distributed exclusively by Leading Edge, a company known for searching out and providing that very thing. Which means that one call to one source can get you any printer, any time you need it, for any purpose. All backed by a full years' warranty from Leading Edge. (Try *that* on any other line of printers.)

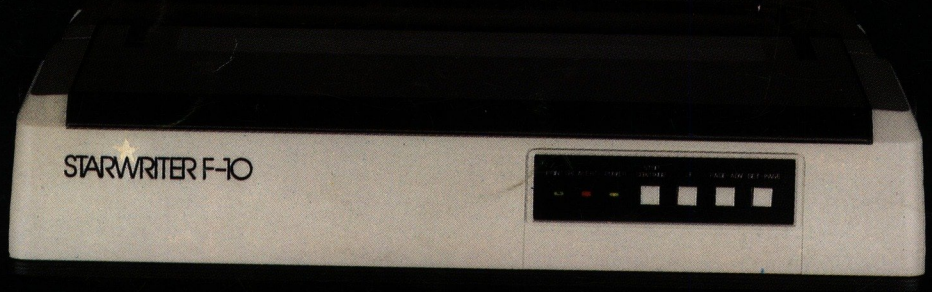
THE PRO'S.

The Prowriters: business printers—and more. The "more" is a dot-matrix process with more dots. It gives you denser, correspondence quality copy (as opposed to business quality copy, which looks like a bad job of spray-painting).
Prowriter: 120 cps. 80 columns dot matrix compressable to 136. 10" carriage. Parallel or serial interface.
Prowriter 2: Same as Prowriter, except 15" carriage allows full 136 columns in normal print mode. Parallel or serial interface.



THE STAR.

The Starwriter F-10. In short (or more precisely, in a sleek 6" high, 30-pound unit), it gives you more of just about everything—except bulk and noise—than any other printer in its price range. It's a 40 cps letter-quality daisy-wheel with a bunch of built-in functions to simplify and speed up word processing. It plugs into almost any micro on the market, serial or parallel.



THE MASTER.

The Printmaster F-10. Does all the same good stuff as the Starwriter except, at 55 cps, the Master does it faster.

Distributed Exclusively by Leading Edge Products, Inc., 225 Turnpike Street, Canton, Massachusetts 02021. Call: toll-free 1-800-343-6833; or in Massachusetts call collect (617) 828-8150. Telex 951-624.